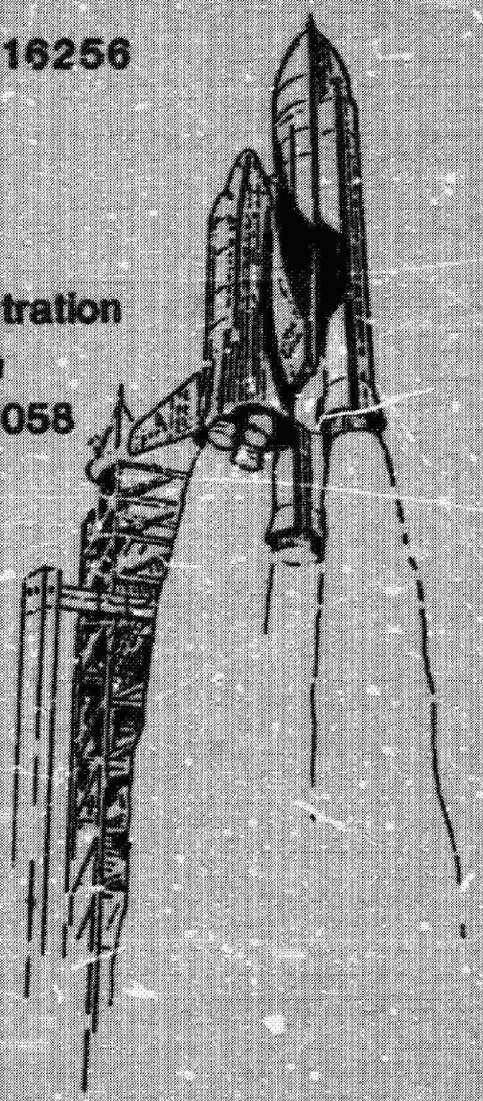
LMSC-HREC TR D867400-II

HIGH ALTITUDE CHEMICALLY REACTING GAS PARTICLE MIXTURES VOLUME II - PROGRAM MANUAL FOR RAMP2

OCTOBER 1984
CONTRACT NAS9-16256

Prepared for

National Aeronautics and Space Administration
Aerodynamic Systems Analysis Section
Johnson Space Center, Houston, TX 77058



by

S. D. Smith

Lockheed
Missiles & Space Company, Inc.
Huntsville Research & Engineering Center
4800 Bradford Drive, Huntsville, AL 35807

FOREWORD

This document is Volume II of a three volume report describing the Reacting and Multi-Phase (RAMP2) computer code developed by the Computational Mechanics Section of Lockheed's Huntsville Research & Engineering Center. Volume II provides a detailed description of all the elements used in the RAMP2 code. Volume I deals with the theory and numerical solution for the computer code, and Volume III is the program users and applications manual.

Documentation of the computer code was prepared in partial fulfillment of Contract NAS9-16256 with the NASA-Lyndon B. Johnson Space Center, Houston, Texas. The contracting officer's technical representative for this study was Mr. Barney B. Roberts (ET41).

The author acknowledges the efforts of Dr. Terry F. Greenwood of NASA-Marshall Space Flight Center and Mr. S.J. Robertson of Lockheed-Huntsville, both of whom contributed to the development of the RAMP2 code.

Companion documents to this report include a theory and numerical solution document for RAMP2 computer code; a user's and applications manual for RAMP2, a report which describes the modifications made to the NASA-Lewis TRAN72 computer code, the original documentation of the NASA-Lewis TRAN 72 computer code, and the original documentation of the Boundary Layer Integral Matrix (BLIMPJ) computer code. These documents are, respectively:

- "High Altitude Supersonic Flow of Chemically Reacting Gas-Particle Mixtures - Volume I - A Theoretical Analysis and Development of the Numerical Solution," LMSC-HREC TR D867400-I.

 I, II

- "High Altitude Supersonic Flow of Chemically Reacting Gas-Particle Mixtures - Volume III - RAMP2 - Computer Code Users and Applications Manual," LMSC-HREC TR D867400-III.

- "User's Guide for TRAN72 Computer Code Modified for Use with RAMP and VOFMOC Flowfield Codes," LMSC-HREC TM D390409.

- Svehla, R.A., and B.J. McBride, "FORTRAN IV Computer Program for Calculation of Thermodynamics and Transport Properties of Complex Chemical Systems," NASA TN D-7056, January 1976.

- Evans, R.M., "Boundary Layer Integral Matrix Procedure BLIMP-J User's Manual," Aerotherm UM-75-64, July 1975.

CONTENTS

1. INTRODUCTION AND SUMMARY

This document, Volume II, provides detailed description of the flowfield element of the high altitude version of the Reacting and Multi-Phase (RAMP2) computer program. The goal in developing RAMP2 was to develop computer code which would provide an engineering tool for calculating high altitude (orbital) rocket exhaust plumes. Additionally, the computer model includes the major controlling phenomena (boundary layer, chemistry, two-phase flow, and free molecular flow) which influence the structure of the plume.

The contents of this volume are not intended to supply the reader with a complete description of the RAMP2. Instead it is intended to give, along with the contents of Volumes I and III (Refs. 1 and 2, respectively), the necessary information about the internal functions of RAMP2 to enable the user to understand and modify the basic coding of this computer program.

RAMP2 consists of three basic computational modules: the TRAN72 program for generating equilibrium thermodynamic and transport data, the RAMP2F code for solving the flowfield and the BLIMPJ code used to calculate the nozzle boundary layer. Due to computer storage limitations the three programs must be executed separately; however, communication between the programs has been provided via temporary files so that except for separate executions they can be considered as one program.

Detailed descriptions of the TRAN72 and BLIMPJ (Refs. 3 and 4, respectively) programs have not been included in this report. Section 3 of this report does contain a brief description of the subroutines of each of the codes. Volume III discusses the use of TRAN72 and BLIMPJ along with RAMP2. Complete descriptions of the two codes are given in Refs. 3 and 4.

Section 2 of this volume describes the overall flow of RAMP2 (including TRAN72 and BLIMPJ) along with the data communicated between the various modules within the RAMP2F flowfield code as well as between RAMP2, TRAN72, and BLIMPJ. The overlay structure (RAMP2F, TRAN72, BLIMPJ) and labeled common blocks used by each subprogram of the RAMP2F flowfield program is presented in Section 3. The files used by RAMP2 are described in Section 4. A complete subroutine by subroutine description of all the codes is included in Section 5. Appendix A lists the overlay control cards for the Univac 1108 and CDC 7600. Appendix B contains discussion on how to convert the program to the CDC 7600.

2. PROGRAM FLOW CHARTS

This section gives the user a general description of the overall flow of the computer codes which make up the RAMP2 code. Additionally, more detail on the flow fo the RAMP2F code is also included. Flow charts for the TRAN72 and BLIMPJ codes can be found in Refs. 3 and 4.

Figure 2-1 presents the basic flow of the three programs which make up the RAMP2 code. First, the TRAN72 program is executed to provide the thermodynamic data for the nozzle and plume solutions. Next, the RAMP2F code is executed to provide the nozzle solution and input data for the boundary layer program (BLIMPJ). Then BLIMPJ is run to generate the nozzle wall boundary layer and exit plane boundary layer properties. Finally, the RAMP2F code is re-executed to merge the inviscid nozzle and boundary layer results and calculate the plume.

Figure 2-2 presents the communication of the various auxiliary programs with the RAMP2 programs along with communication between the three programs of the RAMP2 code.

Figure 2-3 presents a flow chart of the main routines in functional groupings for the RAMP2F program. To attempt to completely flow chart the entire program would probably transmit less information than that given in Fig. 2-3 since it would be extremely complex and bulky.

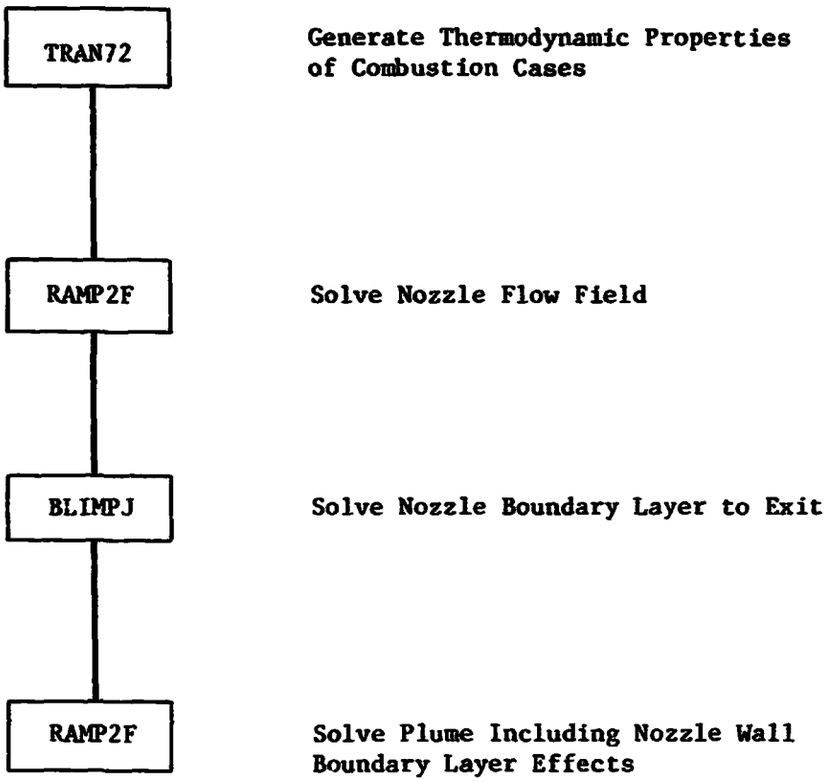


Fig. 2-1 Overall Flow of RAMP2 Programs

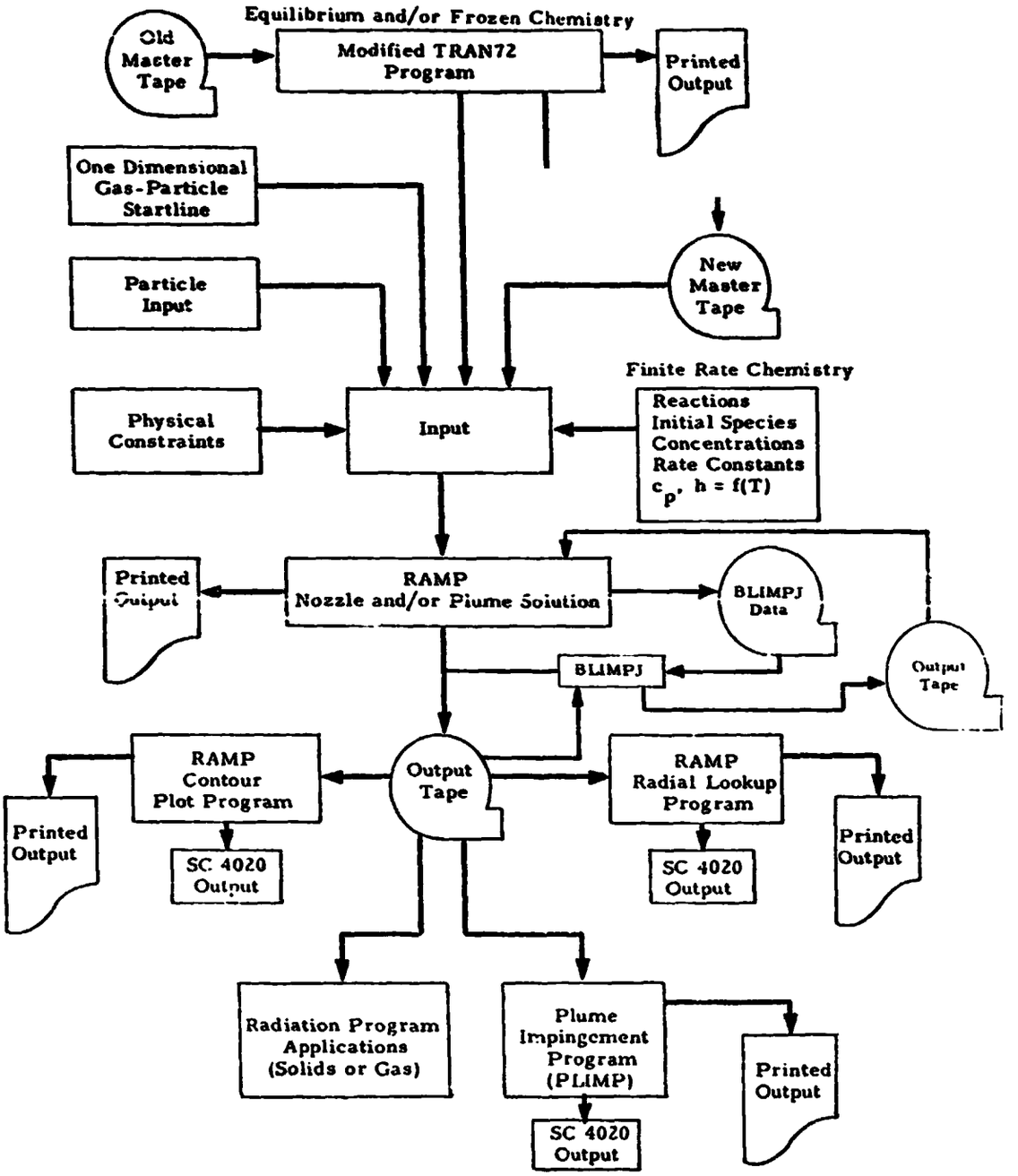


Fig. 2-2 Sequencing and Communication of Auxiliary Programs with the RAMP Program

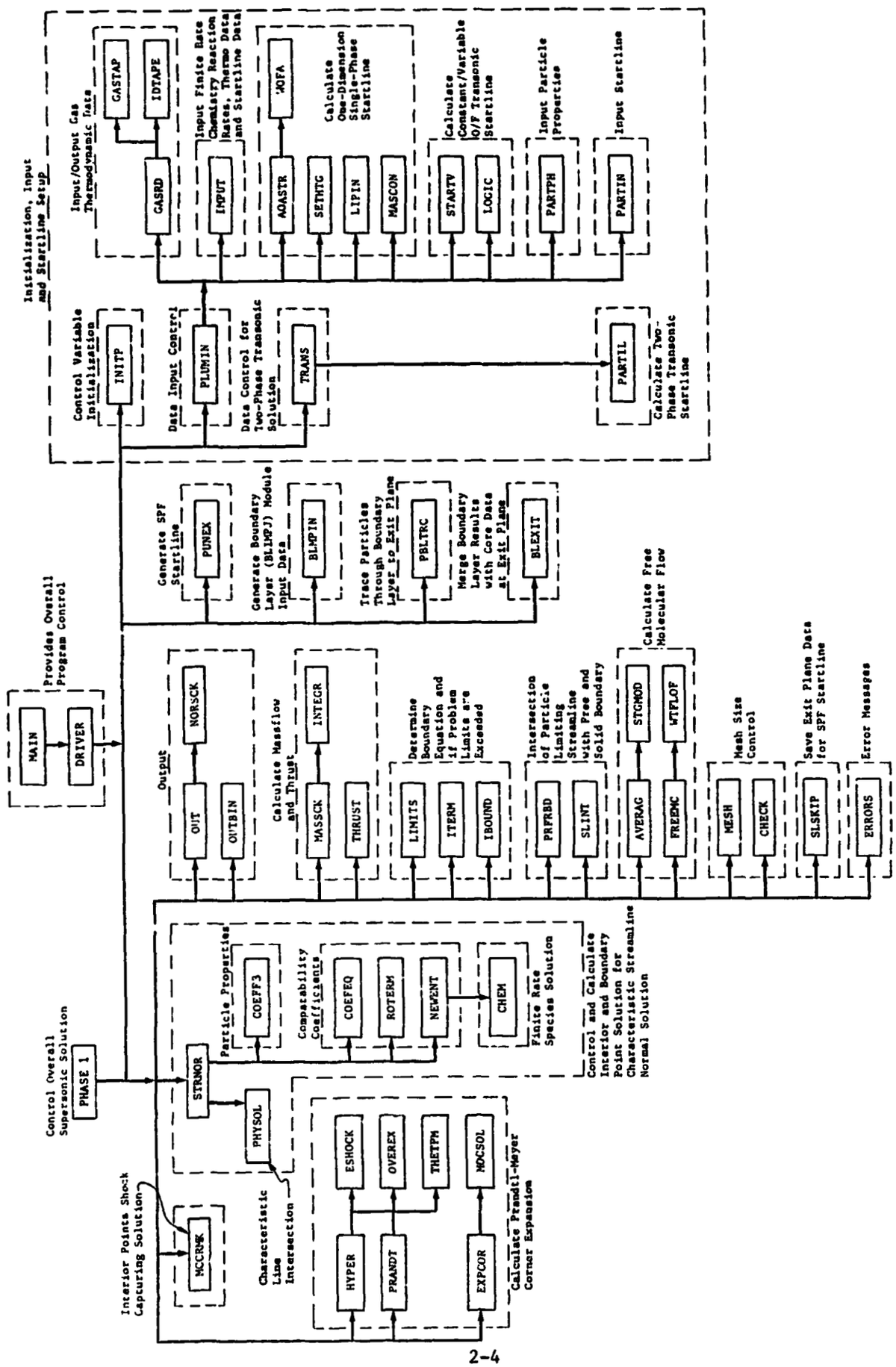


Fig. 2-3 Basic RAMP2F Flow Chart Broken Down into Functional Groupings

3. OVERLAY AND COMMON BLOCK DESCRIPTION

Section 3.1 describes the overlay structure of the RAMP2 codes while Section 3.2 lists the references and cross references of every subroutine and common block in the RAMP2 flowfield code.

3.1 PROGRAM OVERLAY STRUCTURE

Three separate programs (TRAN72, RAMP2F, and BLIMPJ) make up the RAMP2 program. These programs are presently operational on the Univac 1110 and CDC 7600 computer systems. The TRAN72 program does not require an overlay to stay within the core requirements of the Univac 1110 or CDC 7600 computers although it is possible to overlay the program as described in Ref. 3. The RAMP2F flowfield and BLIMPJ codes do require overlaying in order to fit on the Univac 1110 and CDC 7600 systems.

The RAMP2F flowfield program overlay structure is shown in Fig. 3-1 and detailed in Table 3-1. The BLIMPJ program overlay structure is given in Fig. 3-2 and Table 3-2. The list of subroutines for the BLIMPJ code given in Table 3-2 gives the subroutine name followed by the element name in parenthesis. The Univac 1110 overlay loader instructions and CDC 7600 segmentation loader instructions are given in Appendix A.

The amount of storage requirements for each machine depends on the size of the words and efficiency of the compiler for each machine. Typical numbers are given below in decimal words (octal words).

	Univac 1110	CDC 7600 FTN4 Version
TRAN72 Program Size	55964(156725)	(124300)
RAMP2F Program Size (Overlayed)	74230(222726)	(133300)
BLIMPJ Program Size (Overlayed)	62280(173233)	(136300)

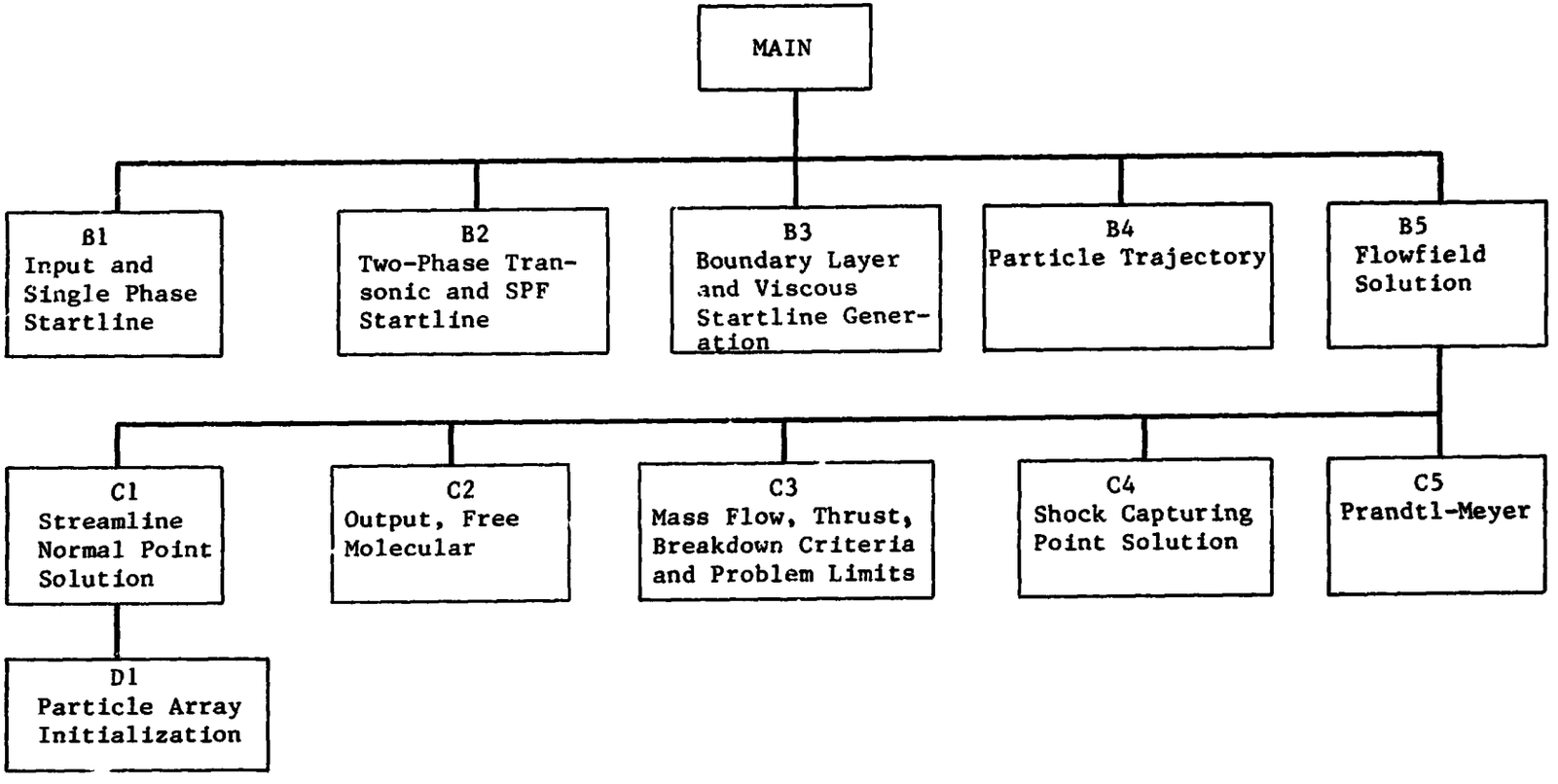


Fig. 3-1 RAMP Overlay Structure

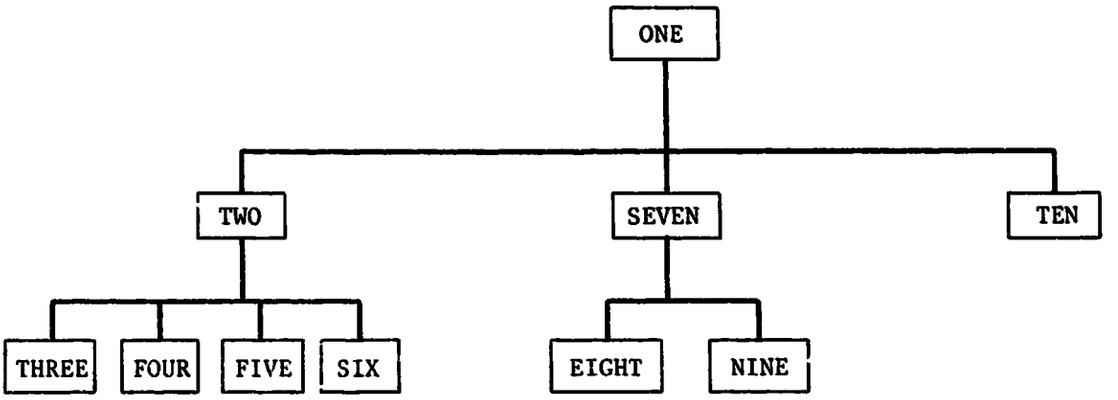


Fig. 3-2 BLIMPJ Overlay Structure

Table 3-1 RAMP OVERLAY

MAIN

DRIVER, THERMO, AND OTHER COMMON USE ROUTINES

ALGINT	IBOUND	REPEAT	THERMV
BLKDAT	IDMPFP	RGVOFM	THERM1
BOUND	ITSUB	RHOPEM	TKEY
DRAGCP	KIKOFF	RITE	TOFEM
DRAGMR	LAGRNG	SITER	TOFH
DRIVER	MAIN	SPCTX	TOFV
EMOFV	PAGE	TAB	UDFEM
ERRORS	PFP	TEMTAB	UOFV
FABLE	POFEM	THERMO	VOFEM
			XSI

B1

INPUT AND SINGLE-PHASE STARTLINE

AOASTR	INITP	PAGVOF	SPACET
AXIS	LIPIN	PARTIN	STARTV
GASRD	LOGIC	PARTPH	STLINE
GASTAP	MASCON	PHI	THERMT
IDMPHI	MASS	PLMOUT	WG
IDMTAB	MAXT	PLUMIN	WOFA
IDMXSI	OFSET	POINT	WTT
IDTAPE	OUTPUT	SETHTG	WXANDR
IMPUT			

B2

TWO-PHASE TRANSONIC AND SPF STARTLINE

ARASSL	HALL	PARTIL	SPECIE
COEFS	IDMPOP	POP	TRACE
DLTA	ONED	PROP	TRANS
FIXIL	ORTHLS	PUNEX	WDGI

Table 3-1 (Continued)

B3

BOUNDARY LAYER INPUT DATA GENERATION

BLMPIN	IDSPEC	SPECIE	WALPRP
IDMTOB	SPECC		

B4

PARTICLE TRAJECTORY TRACING AND MERGING ROUTINES FOR VISCOUS EXIT STARTLINE

BLEXIT	INTERP	PREAD	TAPMOV
GAS	PARLOK	PRO	TRACEP
IDMPDT	PBLTRC	READF	WRITP
IDMPRO	PDT	START	WTFLOP
INRSCP			

B5

FLOWFIELD SOLUTION

CHECK	EMOFP	PHASE1	SOKINT
CHEM	FNEWTN	PRATPT	VEMAG
COEFEQ	GAPPBI	RGMOFP	VMODEL
COEFF3	INRSCT	ROTERM	VMOD1
DOTPRD	NEWENT	SLDP	

C1

STREAMLINE NORMAL POINT SOLUTION

AVERAG	PHYSOL	SOKSOL	STRNOR
CARCTR	SLPLIN	STGMOD	

C2

OUTPUT AND FREE MOLECULAR

DELTAF	ITARM	OUTBIN	VMOD2
FREEMC	LIMITS	POFH	WEAK
ENTROP	NORSCK	TOFENH	WTFLOF
ESHOCK	OUT	TURN	INTT
SLSKIP	PRFRBD		

Table 3-1 (Concluded)

C3

MASS FLOW, THRUST, BREAKDOWN CRITERIA, AND PROBLEM LIMITS

CBREAK	ITERM	MASSCK	ThRUST
INTEGR			

C4

SHOCK CAPTURING POINT SOLUTION AND MISCELLANEOUS

BACWRD	CODEF	FORWRD	CODEE
BOUNDA	CODEH	MCCRMK	DECODE

C5

PRANDTL-MEYER

DELTFF	IRAD	PAFH	PRANDT
ENTRPP	MESH	PARINT	SLINT
ESHOCC	MOCSOL	PARSTR	THETPM
EXPCOR	NORMCK	PLOAD	WEKK
HYPER	OVEREX		

D1

PARTICLE ARRAY INITIALIZATION

DUMSYS

Table 3-2 BLIMPJ OVERLAY

ONE

BLIMP(B2A)	ERP (B30A)	MATS1 (B18A)	SLOPL (B16B)
CHOMO (B14D)	ETHEF (B30B)	MATS2 (B18B)	STATE (B14A)
CRECT (B23A)	HHOMO (B14C)	PROPS (B25A)	TAYLOR (B26A)
DUMCOM (B01A)	KINET (B28A)	RERAY (B15B)	THERM (B21A)
EQUIL (B20A)	LIAD (B30C)	SECOND (B30G)	TRACER (B19B)
ERF (B19B)	MATER (B22A)	SHOMO (B14E)	TRIBL (B19A)

TWO

DATE (B30E)	LINMAT (B27A)
FIRSTG (B29A)	SETUP (B03A)
HISTXI (B10A)	TOD (B30F)

THREE

REFCON (B07A)	SLOPQ (B16A)

FOUR

GEOM (B09B)	RECASE (P09A)
MISCIN (B07B)	

FIVE

INPUT (B24A)

SIX

STATEN (B14B)

SEVEN

ABMAX (B17A)	ITERAT (B04A)	OGLE (B36A)
ICOEFF (B08B)	LINCER (B06A)	RNLCER (B05C)
IMONE (B12B)	NNNCER (B05B)	TLEFT (B30D)
IONLY (B13B)		

Table 3-2 (Concluded)

EIGHT

FILQ3 (B50A)	FISLEQ (B50D)	POINTS (B50F)
FILQ5 (B50B)	FUNXS (B50E)	REFIT (B50?)
FINEQ (B50C)	OUTBL (B11A)	TRINT (B50H)

NINE

ROCOUT (B11B)

TEN

EDGHSP (EDGHSP)	READP (READP)
EDGPRP (EDGPRP)	TAD (TAD)
IDMTAD (IDMTAD)	TAPMOV (TAPMOV)

3.2 SUBROUTINE AND COMMON BLOCK REFERENCES

This subsection provides a list of all the references and cross-references for the subroutines and common blocks in the RAMP2F code. The same information for the BLIMPJ and TRAN72 codes is not included in this document, however, the user can refer to Refs. 3 and 4 to find similar information. Table 3-3 provides a list of all the subroutines in the RAMP2F program, along with each routine's entry point(s) and the subroutines called by each routine. Table 3-4 is a cross reference of Table 3-3. Table 3-4 provides a list of all entry points, the element (subroutine) that it is part of along with a list of all elements (subroutines) which reference the entry point.

Table 3-5 lists each common block in the RAMP2F program along with its size (decimal) and routines that refer to each common block. This table can be used to locate all references to a particular common block in the event of a program modification which requires a length change to the common block.

Table 3-3 RAMP2F ENTRY POINTS AND EXTERNAL REFERENCES

Element/ Version	Entry Points	External References
ALGINT	*ALGINT	
AOASTR	*AOASTR	ERRORS, ITSUB, RGFPVM, THERMO, WOFA
ARASSL	*ARASSL	
SVERAG	*AVERAG	EMOFV, POFEM, RHOFEM, STGMOD, WOFA
AXIS	*AXIS	IDMPHI, PHI
BACWRD	*BACWRD	CODEE, CODEF
BLEXIT	*BLEXIT	IDMPOB, IDMPFP, INRSCP, ITSUB, PDT, PFP, POFEM, PREAD, RGVOFM, SITER, TOFEM, UOFEM
BLKDAT		
BLMPIN	*BLMPIN	IDMTDG, IDSPEC, SPECIB, TAB, WALPRP
BOUND	*BOUND	LAGRNG
BOUNDA	*BOUNDA	BOUND, INRSCT
CARCTR	*CARCTR	
CBREAK	*CBREAK	THERMO
CHECK	*CHECK	CAPPBI, IDMPFP, PFP, SPCTX, VMODEL
CHEM	*CHEM	SLDP, TKEY
CODEE	*CODEE	PFP
CODEF	*CODEF	PFP
CODEH	*CODEH	PFP, PPATPT
COEFEQ	*COEFEQ	

Table 3-3 (Continued)

Element/ Version	Entry Points	External References
COEFF3	*COEFF3	IDMPFP, PFP
COEFS	*COEFS	
DECODE	*DECODE	EMOFV, IDMPFP, THERMO TOFEM, UOFEM
DELTAF	*DELTAF	
DLTA	*DLTA	PROP
DOTPRD	*DOTPRD	
DRAGCP	*DRAGCP	
DRAGMR	*DRAGMR	ALGINT
DRIVER	*DRIVER	BLEXIT, BLMPIN, INITP, PBLTRC, PHASE1, PLUMIN, PUNEX, SPACET, TRANS
DUMSYS	*DUMSYS	
EMOFP	*EMOFP	
EMOFV	*EMOFV	TOFV
ENTROP	*ENTROP	
ENTRPP	*ENTRPPP	
ERRORS	*ERRORS	REPEAT
ESHOCC	*ESHOCC	DELTFF, EMOFV, ENTRPP, POFEM, RHOFEM, THERMO, WEKK
ESHOCK	*ESHOCK	DELTAF, EMOFV, ENTROP, POFEM, RHOFEM, THERMO, WEAK
EXPCOR	*EXPCOR	ALGINT, CHECK, INRSCT, MOCSOL, PPATPT, SPCTX, VMODEL
FABLE	*FABLE	EMOFV, POFEM, TAB, TOFV, XSI
FIXIL	*FIXIL	PROP

Table 3-3 (Continued)

Element/ Version	Entry Points	External References
FNEWTN	*FNEWTN	
FORWRD	*FORWRD	CODE, CODF
FREEMC	*FREEMC	BOUND, IDMPFP, INRSCT, ITARM, OUT, OUTBIN, PFP, WTFLOF
GAPPBI	*GAPPBI	ALGINT, DRAGCP, DRAGMR, EMOFV, PFP, POFEM, STRNOR, TEMTAB, THERMO, TOFV, UOFV
GAS	*GAS	PRO
GASRD	*GASRD	GASTAP, IDMXSI, IDTAPE, TAB, XSI
GASTAP	*GASTAP	ERRORS, IDMTAB, INPUT
HALL	*HALL	
HYPER	*HYPER	EMOFV, ERRORs, FNEWTN, ITSUB, OVEREX, PAFH, POFEM, THERMO, THETPM, TOFH
IBOUND	*IBOUND	
IDMPDT	*IDMPDT	
IDMPFP	*IDMPFP	
IDMPHI	*IDMPHI	
IDMPOP	*IDMPOP	
IDMPRO	*IDMPRO	
IDMTAB	*IDMTAB	
IDMTOB	*IDMTOB	
IDMXSI	*IDMXSI	
IDSPEC	*IDSPEC	
IDTAPE	*IDTAPE	TAB

Table 3-3 (Continued)

Element/ Version	Entry Points	External References
IMPUT	*IMPUT	SPCTX
INITP	*INITP	
INRSCP	*INRSCP	ERRORS
INRSCT	*INRSCT	ERRORS
INTEGR	*INTEGR	PFP, VEMAG
INTERP	*INTERP	PFP
INTT	*INTT	IDMPFP, OUT, PFP, PPATPT
IRAD	*IRAD	SPCTX
ITARM	*ITARM	
ITERM	*ITERM	
ITSUB	*ITSUB	
KIKOFF	*KIKOFF	
LAGRNG	*LAGRNG	
LIMITS	*LIMITS	BOUND
LIPIN	*LIPIN	RGVOFM, THERMO, UOFV
LOGIC	*LOGIC	AXIS, BOUND, IBOUND, IDMPHI, OUTPUT, PHI, POINT, SITER, THERMT
MAIN	*MAIN	DRIVER, DUMSYS
MASCON	*MASCON	EMOFV, ERRORS, ITSUB, RGVOFM, RHOFEM
MASS	*MASS	PHI, THERMT
MASSCK	*MASSCK	IDMPFP, INTEGR, PFP
MAXT	*MAXT	

Table 3-3 (Continued)

Element/ Version	Entry Points	External References
MCCRMK	*MCCRMK	BACWRD, CODEE, CODEF, CODEH, DECODE, FORWRD, IDMPFP, PFP
MESH	*MESH	EMOFV, THERMO
MOCSOL	*MOCSOL	ALGINT, BOUND, COEFEQ, COEFF3, ERRORS, FNEWTN, IDMPFP, INRSCT, NEWENT, PFP, PPATPT, RGMOFP, ROTERM, SPECTX, THERM1, VMODEL, VOFEM
NEWENT	*NEWENT	CHEM
MORMCK	*NORMCK	INRSCT
NORSCK	*NORSCK	ITSUB, TOFENH
OFSET	*OFSET	
ONED	*ONED	
ORTHLS	*ORTHLS	
OUT	*OUT	ESHOCK, NORSCK, PAGE, PFP, POFEM, PPATPT, SPCTX, THERMO, VEMAG, VMOD1, VMOD2
OUTBIN	*OUTBIN	EMOFV, PFP, SPCTX, TEMTAB, THERMO, VMOD1
OUTPUT	*OUTPUT	MASS, PAGVOF, PHI, THERMT
OVEREX	*OVEREX	EMOFV, ERRORS, ESHOCC, IDMPFP, ITSUB, PFP, POFEM, THERMO, UOFV
PAFH	*PAHF	TOFH
PAGE	*PAGE	
PAGVOF	*PAGVOF	
PARINT	*PARINT	CHECK, IDMPFP, INRSCT, PFP

Table 3-3 (Continued)

Element/ Version	Entry Points	External References
PARLOK	*PARLOK	INTERP, READF, TAPMOV
PARSTR	*PARSTR	PFP
PARTIL	*PARTIL	COEFS, DLTA, EMOFV, FIXIL, ONED, ORTHLS, PAGE, PROP, SITER, TAB, THERMO, TOFEM, TOFV, TRACE, WDGI
PARTIN	*PARTIN	IDMPFP, PFP, POFEM, RGVOFM, SPCTX, THERMO, TOFEM, UOFEM
PARTPH	*PARTPH	PAGE
PBLTRC	*PBLTRC	IDMPDT, IDMPRO, INRSCP, PARLOK, PRO, START, TAPMOV, TRACEP, WRITP, WRFLOP
PDT	*PDT	
PFP	*PFP	
PHASE1	*PHASE1	ALGINT, AVERAB, BOUND, BOUNDA, CVREAK, CHECK, EMOFV, ERRORS, EXPCOR, FREEMC, HYPER, IBOUND, IDMPFP, INRSCT, INTT, IRAD, ITERM, LIMITS, MASSCK, MCCRMK, MESH, NORMCK, OUT, OUTBIN, PARINT, PARSTR, PFP, PLOAD, POFEM, PPATPT, PRANDT, PRFRBD, RGMOFP, SLINT, SLSKIP, SOKINT, SOKSOL, SPCTX, STRNOR, THERMO, THETPM, THRUST, TOFEM, TURN, UOFV, VMODEL, VMOD1, VOFEM
PHI	*PHI	INRSCT, PFP, THERMO, UOFV
PHYSOL	*PHYSOL	BOUND, GAPPBI, INRSCT, ITSUB, PFP, PPATPT, THERMO, UOFV, VMODEL
PLMOUT	*PLMOUT	EMOFV, IDMTAB, PAGE, PFP, TAB, THERMO
PLOAD	*PLOAD	IDMPFP, PFP, SPCTX

Table 3-3 (Continued)

Element/ Version	Entry Points	External References
PLUMIN	*PLUMIN	AOASTR, BOUND, GASRD, IBOUND, LIPIN, LOGIC, MASCON, PARTIN, PARTPH, PLMOUT, SETHTG, STARTV, STLINE
POFEM	*POFEM	
POFH	*POFH	TOFH
POINT	*POINT	IDMPHI, PHI, THERMT, WG, STT, WXANDR
POP	*POP	
PPATDP	*PPATPT	DRAGCP, DRAGMR, EMOFV, PFP, POFEM, TEMTAB, THERMO TOFV
PRANDT	*PRANDT	EMOFV, PAFH, POFEM, SPCTX, THERMO, THETPM, TOFH, TOFV, UOFV, VMODEL
PREAD	*PREAD	IDMPFP
PRFRBD	*PRFRBD	IDMPFP, INRSCT, PFP, PPATPT
PROG	*PRO	
PROP	*PROP	HALL
PUNEX	*PUNEX	ARASSL, IDMPOP, SPECIE
READF	*READF	IDMPFP
REPEAT	*REPEAT	
RGMOFP	*RGMOFP	EMOFP, EMOFV, ERRORS, ITSUB, POFEM, THERMO, VOFEM
RGVOFM	*RGVOFM	EMOFV, ERRORS, ITSUB, TAB, THERMO, VOFEM
RHOFEM	*RHOFEM	POFEM
RITE	*RITE	
ROTERM	*ROTERM	

Table 3-^ (Continued)

Element/ Version	Entry Points	External References
SETHTG	*SETHTG	THERMO, TKEY
SITER	*SITER	EMOFV, ITSUB, POFEM, RGVOFM, THERMO
SLDP	*SLDP	
SLINT	*SLINT	PPATPT, SPCTX, UOFEM
SLPLIN	*SLPLIN	
SLSKIP	*SLSKIP	PFP, SPCTX, TEMTAB
SOKINT	*SOKINT	
SOKSOL	*SOKSOL	
SPACET	*SPACET	
SPCTX	*SPCTX	
SPECC	*SPECC	
SPECIB	*SPECIB	SPECC, TAB
?ECIE	*SPECIE	POP, TAB, THERMO
.TART	*START	
STARTV	*STARTV	AOASIR, BOUND, IBOUND, IIMPHI, ITSUB, MASS, MAXT, OFSET, OUTPUT, PHI, POFEM, RGVOFM, RHOFEM, THERMT
STGMOD	*STGMOD	
STLINE	*STLINE	
STRNOR	*STRNOR	BOUND, CHECK, COEFEQ, COEFF3, EMOFV, FNEWTN, GAPPB IDMPFP, INRSCT, NEWENT, PFP, PHYSOL, PPATPT, RGMOFP, ROTERM, SITER, SLPLIN, SPCTX, UOFV, VMODEL, VOFEM

Table 3-3 (Continued)

Element/ Version	Entry Points	External References
TAB	*TAB	
TAPMOV	*TAPMOV	
TEMTAB	*TEMTAB	
THERMO	*THERMO	FABLE, TAB, THERMV, THERM1
THERMT	*THERMT	
THERMV	*THERMV	EMOFV, POFEM, TOFV
THERM1	*THERM1	TKEY, TOFH
THETPM	*THETPM	ERRORS, ITSUB, THERMO, TOFH, TOFV
THRUST	*THRUST	PFP, VEMAG
TKEY	*TKEY	
TOFEM	*TOFEM	
TOFENH	*TOFENH	ITSUB, TKEY
TOFH	*TOFH	ITSUB, TKEY
TOFV	*TOFV	ERRORS, KIKOFF, RITE
TRACE	*TRACE	PROP
TRACEP	*TRACEP	DRAGCP, DRAGMR, GAS, TEMTAB
TRANS	*TRANS	PARTIL, TAB, THERMO
TURN	*TURN	EMOFV, ERRORS, ESHOCK, ITSUB, THERMO, UOFEM, UOFV
UOFEM	*UOFEM	ERRORS, KIKOFF, RITE
UOFV	*UOFV	EMOFV, UOFEM

Table 3-3 (Concluded)

Element/ Version	Entry Points	External References
VEMAG	*VEMAG	DOTPRD
VMODEL	*VMODEL	
VMOD1	*VMOD1	
VMOD2	*VMOD2	
VOGEM	*VOFEM	TOFEM
WALPRP	*WALPRP	
WDGI	*WDGI	PROP
WEAK	*WEAK	DELTAF, EMOFV, ENTROP, POFEM, RHOFEM, THERMO
WEKK	*WEKK	DELTFF, EMOFV, ENTRPP, POFEM, RHOFEM, THERMO
WG	*WG	
WOFA	*WOFA	
WRITP	*WRIPT	IDMPDT, PDT
WTFLOF	*WTFLOF	IDMPDT, PDT
WTT	*WTT	
WXANDR	*WXANDR	PHI
XSI	*XSI	

Table 3-4 -- RAMP2F ENTRY POINT REFERENCES

Entry Points	Element	Elements Referencing
*ALGINT	ALGINT	DRAGMR, EXPCOR, GAPPBI, MOCSOL, PHASE1
*AOASTR	AOASTR	PLUMIN, STARTV
*ARASSL	ARASSL	PUMEX
*AVERAG	AVERAG	PHASE1
*AXIS	AXIS	LOGIC
*BACWRD	BACWRD	MCCRMK
*BLEXIT	BLEXIT	DRIVER
*BLMPIN	BLMPIN	DRIVER
*BOUND	BOUND	BOUNDA, FREEMC, LIMITS, LOGIC, MOCSOL, PHASE1, PHYSOL, PLUMIN, STARTV, STRNOR
*BOUNDA	BOUNDA	PHASE1, PHYZOL
*CARCTR	CARCTR	PHYZOL
*CBREAK	CBREAK	PHASE1
*CHECK	CHECK	EXPCOR, PARINT, PHASE1, STRNOR
*CHEM	CHEM	NEWENT
*CODEE	CODEE	BACWRD, FORWRD, MCCRMK
*CODEF	CODEF	BACKWRD, FORWRD, MCCRMK
*CODEH	CODEH	MCCRMK
*COEFEQ	COEFEQ	MOCSOL, STRNOR
*COEFF3	COEFF3	MOCSOL, STRNOR

Table 3-4 (Continued)

Entry Point	Element	Elements Referencing
*COEFS	COEFS	PARTIL
*DECODE	DECODE	MCCRMK
*DELTAF	DELTAF	ESHOCK, WEAK
*DELTFF	DELTFF	ESHOCC, WEKK
*DLTA	DLTA	PARTIL
*DOTPRD	DOTPRD	VEMAG
*DRAGCP	DRAGCP	GAPPBI, PPATPT, TRACEP
*DRAGMR	DRAGMR	GAPPBI, PPATPT, TRACEP
*DRIVER	DRIVER	MAIN
*DUMSYS	DUMSYS	MAIN
*EMOFP	EMOFP	RGMOFP
*EMOFV	EMOFV	AVERAG, DECODE, ESHOCC, ESHOCK, FABLE, GAPPBI, HYPER, MASCON, MESH, OUTBIN, OVEREX, PARTIL, PHASE1, PLMOUT, PPATPT, PRANDT, RGMOFP, RFVOFM, SITER, STRNOR, THERMV, TURN, UOFV, WEAK, WEKK
*ENTROP	ENTROP	ESHOCK, WEAK
*ENTRPP	ENTRPP	ESHOCC, WEKK
*ERRORS	ERRORS	AOASTR, BOUNDA, GASTAP, HYPER, INRSCP, INRSCT, MASCON, MOCSOL, OVEREX, PHASE1, PHYZOL, RGMOFP, RGVOFM, SOKFLX, THETPM, TOFV, TURN, UOFEM
*ESHOCC	ESHOCC	OVEREX
*ESHOCK	ESHOCK	OUT, SOKFLX, TURN

Table 3-4 (Continued)

Entry Point	Element	Elements Referencing
*EXPCOR	EXPCOR	PHASE1
*FABLE	FABLE	THERMO
*FIXIL	FIXIL	PARTIL
*FNEWTN	FNEWTN	HYPER, MOCSOL, STRNOR
*FORWRD	FORWRD	MCCRMK
*FREEMC	FREEMC	PHASE1
*GAPPBI	GAPPBI	CHECK, COEFF3, PHYSOL, STRNOR
*GAS	GAS	TRACEP
*GASRD	GASRD	PLUMIN
*GASTAP	GASTAP	GASRD
*HALL	HALL	PROP
*HYPER	HYPER	PHASE1
*IBOUND	IBOUND	LOGIC, PHASE1, PLUMIN, STARTV
*IDMPDT	IDMPDT	BLEXIT, PBLTRC, WRITP, WTFLOP
*IDMPFP	IDMPFP	BLEXIT, CHECK, COEFF3, DECODE, FREEMC, INTT, MASSCK, MCCRMK, MOCSOL, OVEREX, PARINT, PARTIN, PHASE1, PLOAD, PREAD, PRFRBD, READF, STRNOR
*IDMPHI	IDMPHI	AXIS, LOGIC, POINT, STARTV
*IDMPOP	IDMPOP	PUNEX
*IDMPRO	IDMPRO	PBLTRC
*IDMTAB	IDMTAB	GASRD, GASTAP, PLMOUT
*IDMTOB	IDMTOB	BLMPIN

Table 3-4 (Continued)

Entry Point	Element	Elements Referencing
*IDMXSI	IDMXSI	GASRD
*IDSPEC	IDSPEC	BLMPIN
*IDTAPE	IDTAPE	GASRD
*IMPUT	IMPUT	GASTAP
*INITP	INITP	DRIVER
*INRSCP	INRSCP	BLEXIT, PBLTRC
*INRSCT	INRSCT	BOUNDA, COEFF3, EXPCOR, FREEMC, MOCSOL, NORMCK, PARINT, PHASE1, PHYSOL, PHYZOL, PRFRBD, STRNOR
*INTEGR	INTEGR	MASSCK
*INTERP	INTERP	PARLOK
*INTT		PHASE1
*IRAD	IRAD	PHASE1
*ITARM	ITARM	FREEMC
*ITERM	ITERM	PHASE1
*ITSUB	ITSUB	ADASTR, BLEXIT, HYPER, MASCON, NORSCK, OVEREX, PHYSOL, RGMOFP, RGVOFM, SITER, SOKFLX, STARTV, THETPM, TOFENH, TOFH, TURN
*KIKOFF	KIKOFF	BOUNDA, TOFV, UOFEM
*LAGRNG	LAGRNG	BOUND
*LIMITS	LIMITS	PHASE1
*LIPIN	LIPIN	PLUMIN

Table 3-4 (Continued)

Entry Point	Element	Elements Referencing
*LOGIC	LOGIC	PLUMIN
*MASCON	MASCON	PLUMIN
*MASS	MASS	OUTPUT. STARTV
*MASSCK	MASSCK	PHASE1
*MAXT	MAXT	STARTV
*MCCRMK	MCCRMK	PHASE1
*MESH	MESH	PHASE1
*MOCSOL	MOCSOL	EXPCOR
*NEWENT	NEWENT	MOCSOL, STRNOR
*NORMCK	NORMCK	PHASE1
*NORSCK	NORSCK	OUT
*OFSET	OFSET	STARTV
*ONED	ONED	PARTIL
*ORTHLS	ORTHLS	PARTIL
*OUT	OUT	FREEMC, INTT, PHASE1
*OUTBIN	OUTBIN	FREEMC, PHASE1
*OUTPUT	OUTPUT	LOGIC, STARTV
*OVEREX	OVEREX	HYPER
*PAFH	PAFH	HYPER, PRANDT
*PAGE	PAGE	OUT, PARTIL, PARTPH, PLMOUT
*PAGVOF	PAGVOF	OUTPUT

Table 3-4 (Continued)

Entry Point	Element	Elements Referencing
*PARINT	PARINT	PHASE1
*PARLOK	PARLOK	PBLTRC
*PARSTR	PARSTR	PHASE1
*PARTIL	PARTIL	TRANS
*PARTIN	PARTIN	PLUMIN
*PARTPH	PARTPH	PLUMIN
*PBLTRC	PBLTRC	DRIVER
*PDT	PDT	BLEXIT, WRITP, WTFLOP
*PFP	PFP	BLEXIT, CHECK, CODEE, CODEF, CODEH, COEFF3, FREEMC, GAPPBI, INTEGR, INTERP, INTT, MASSCK, MCCRMK, MOCSOL, OUT, OUTBIN, OVEREX, PARINT, PARSTR, PARTIN, PHASE1, PHYSOL, PLMOUT, PLOAD, PPATPT, PRFRBD, REDIST, SLSKIP, STRNOR, THRUST
*PHASE1	PHASE1	DRIVER
*PHI	PHI	AXIS, LOGIC, MASS, OUTPUT, POINT, STARTV, WXANDR
*PHYSOL	PHYSOL	STRNOR
*PLMOUT	PLMOUT	PLUMIN
*PLOAD	PLOAD	PHASE1
*PLUMIN	PLUMIN	DRIVER
*POFEM	POFEM	AVERAG, BLEXIT, ESHOCC, ESHOCK, FABLE, GAPPBI, HYPER, OUT, OVEREX, PARTIN, PHASE1, PPATPT, PRANDT, RGMOFP, RHOFEM, SITER, STARTV, THERMV, WEAK, WEKK

Table 3-4 (Continued)

Entry Point	Element	Elements Referencing
*POINT	POINT	LOGIC
*POP	POP	SPECIE
*PPATPT	PPATPT	CODEH, COEFF3, EXPCOR, FREPRO, INTT, MOCSOL, OUT, PHASE1, PHYSOL, PRFRBD, SLINT, STRNOR
*PRANDT	PRANDT	PHASE1
*PREAD	PREAD	BLEXIT
*PRFRBD	PRFRBD	PHASE1
*PRO	PRO	GAS, PBLTRC
*PROP	PROP	DLTA, FIXIL, PARTIL, TRACE, WDGI
*PUNEX	PUNEX	DRIVER
*READF	READF	PARLOK
*REPEAT	REPEAT	ERRORS
*RGMOFP	RGMOFP	MOCSOL, PHASE1, STRNOR
*RGVOFM	RGVOFM	AOASTR, BLEXIT, LIPIN, MASCON, PARTIN, SITER, STARTV
*RHOFEM	RHOFEM	AVERAG, ESHOCC, ESHOCK, MASCON, STARTV, WEAK, WEKK
*RITE	RITE	TOFV, UOFEM
*ROTERM	ROTERM	MOCSOL, STRNOR
*SETHTG	SETHTG	PLUMIN

Table 3-4 (Continued)

Entry Point	Element	Elements Referencing
*SITER	SITER	BLEXIT, LOGIC, PARTIL, STRNOR
*SLDP	SLDP	CHEM
*SLINT	SLINT	PHASE1
*SLPLIN	SLPLIN	STRNOR
*SLSKIP	SLSKIP	PHASE1
*SOKINT	SOKINT	PHASE1
*SOKSOL	SOKSOL	PHASE1
*SPACET	SPACET	DRIVER
*SPCTX	SPCTX	CHECK, EXPCOR, IMPUT, IRAD, MOCSOL, OUT, OUTBIN, PARTIN, PHASE1, PLOAD, PRANDT, SLINT, SLSKIP, STRNOR
*SPECC	SPECC	SPECIB
*SPECIB	SPECIB	BLMPIN
*SPECIE	SPECIE	PUNEX
*START	START	PBLTRC
*STARTV	STARTV	PLUMIN
*STGMOD	STGMOD	AVERAG, FREPRO
*STLINE	STLINE	PLUMIN
*STRNOR	STRNOR	PHASE1
*TAB	TAB	FABLE, GASRD, IDTAPE, PARTIL, PLMOUT, RGVOFM, SPECIE, THERMO, TRANS
*TAPMOV	TAPMOV	PARLOK, PBLTRC

Table 3-4 (Continued)

Entry Point	Element	Elements Referencing
*TEMTAB	TEMTAB	GAPPBI, OUTBIN, PPATPT, SLSKIP, TRACEP
*THERMO	THERMO	AOASTR, AVERAG, CBREAK, DECODE, ESHOCC, ESHOCK, GAPPBI, HYPER, LIPIN, MESH, OUT, OUTBIN, OVEREX, PARTIL, PARTIN, PHASE1, PHYSOL, PHYZOL, PLMOUT PPATPT, PRANDT, RGMOFP, RGVOFM, SETHTG, SITER, SOKFLX, SPECIE, THETPM, TRANS, TURN, WEAK, WEKK
*THERMT	THERMT	LOGIC, MASS, OUTPUT, POINT, STARTV
*THERMV	THERMV	THERMO
*THERM1	THERM1	MOCSOL, THERMO
*THETPM	THETPM	HYPER, PHASE1, PRANDT
*THRUST	THRUST	PHASE1
*TKEY	TKEY	CHEM, SETHTG, THERM1, TOFENH, TOFH
*TOB	TOB	BLMPIN, SPECIB
*TOFEM	TOFEM	AVERAG, BLEXIT, DECODE, PARTIL, PARTIN, PHASE1, VOFEM
*TOFENH	TOFENH	NORSCK
*TOFH	TOFH	HYPER, PAFH, POFH, PRANDT, THERM1, THETPM
*TOFV	TOFV	AVERAG, EMOFV, FABLE, GAPPBI, PARTIL, PPATPT, PRANDT, THERMV, THETPM
*TRACE	TRACE	PARTIL
*TRACEP	TRACEP	PBLTRC
*TRANS	TRANS	DRIVER
*TURN	TURN	PHASE1

Table 3-4 (Concluded)

Entry Point	Element	Elements Referencing
*UOFEM	UOFEM	BLEXIT, DECODE, PARTIN, SLINT, TURN, UOFV
*UOFV	UOFV	GAPPBI, LIPIN, OVEREX, PHASE1, PHYSOL, PHYZOL, PRANDT, SOKFLX, STRNOR, TURN
*VEMAG	VEMAG	INTEGR, OUT, THRUST
*VMODEL	VMODEL	CHECK, EXPCOR, MOCSOL, PHASE1, PHYSOL, PRANDT, STRNOR
*VMOD1	VMOD1	OUT, OUTBIN, PHASE1
*VMOD2	VMOD2	OUT
*VOFEM	VOFEM	MOCSOL, PHASE1, RGMOFP, RGVOFM, STRNOR
*WALPRP	WALPRP	BLMPIN
*WDGI	WDGI	PARTIL
*WEAK	WEAK	ESHOCK
*WEKK	WEKK	ESHOCC
*WG	WG	POINT
*WOFA	WOFA	ADASTR
*WRITP	WRITP	PBLTRC
*WTFLOF	WTFLOF	FREEMC
*WTFLOP	WTFLOP	PBLTRC
*WTT	WTT	POINT
*WXANDR	WXANDR	POINT
*XSI	XSI	FABLE, GASRD

Table 3-5 RAMP2F COMMON BLOCK SIZE AND REFERENCES

Common Block	Size (Decimal)	Routines Referencing
ACOM	1	DRIVER, PLUMIN, WTFLOP
AMF	25	OUT, OUTBIN, PHASE1
AUX	1	DRIVER, EXPCOR, OUTBIN
AVPROP	24	COEFEQ, MOCSOL, NEWENT, STRNOR
AVPRP2	7	COEFEQ, STRNOR
BLMDAT	50	BLMPIN, CHEM, PHASE1
BOMOUT	2	OUT, PARTIN, TURN, UOFEM, UOFV
BPRESW	6	BLEXIT, GASTAP, IDTAPE, PHASE1, PLUMIN
CAPUR	1	BLEXIT, DRIVER, EXPCOR, GAPPBI, LOGIC, PHASE1, PHYSOL, PLUMIN, PUNEX, STRNOR PPATPT
CHEMCN	8	BLEXIT, CHECK, CHEM, COEFEQ, GASTAP, IMPUT, NEWENT, OUT, OUTBIN, PHASE1, PLOAD, PLUMIN, PUNEX, READF, RGVOFM, SETHTG, SLSKIP, SPCTX, STRNOR, THERM1, TOFENH, TOFH
CHEMXX	459	BACWRD, CHECK, CHEM, CODEE, CODEF, CODEH, COEFEQ, DECODE, EXPCOR, FORWRD, IMPUT, MCCRMK, MOCSOL, NEWENT, OUT, OUTBIN PARTIN, PHASE1, PHYSOL, PRANDT, SETHTG, SLSKIP, SPCTX, STRNOR, THERM1, TOFENH, TOFH
CHEMXY	2	NEWENT, STRNOR
CHEMYY	160	CHEM
CHEQ	1	DECODE, DRIVER, GASRD

Table 3-5 (Continued)

Common Block	Size (Decmial)	Routine Referencing
CHOCK	1	GAS, PBLTRC, TRACEP
CONTRL	26	ARASSL, AVERAG, BACWRD, BLEXIT, BLMPIN, BOUNDA, CBREAK, CHECK, CHEM, CODEE, CODEF, CODEH, COEFEQ, COEFF3, DECODE, DRIVER, ERRORS, ESHOCC, ESHOCK, EXPCOR, FABLE, FORWRD, FREEMC, GAPPBI, GASRD, GASTAP, HYPER, IBOUND, IDTAPE, IMPUT, INITP, INTEGR, IRAD, KIKOFF, LAGRNG, LIMITS, LIPIN, MASCON, MASSCK, MCCPMK, MOCSOL, NEWENT, OUT, OUTBIN, PAFH, PAGE, PARTIL, PARTIN, PARTPH, PBLTRC, PHASE1, PHYSOL, PHYZOL, PLMOUT, PLOAD, PLUMIN, POFH, PPATPT, PRANDT, PREAD, PRFRBD, PUNEX, READF, RGMOFP, RGVOFM, SLINT, SLSKIP, SOKFLX, SPCTX, SPECIB, SPECIE, STARTV, STLINE, STRNOR, TAPMOV, TEMTAB, THERMO, THERMV, THERM1, THETPM, THRUST, TRANS, TURN, WTFLOF, WTFLOP
CONVV	5	PBLTRC, READF, TRACEP, WRITP
CPMUK	3	AVERAG, BLMPIN, CBREAK, COEFF3, DECODE, DRIVER, FABLE, GAPPBI, IMPUT, MOCSOL, PAFH, POFH, PPATPT, PRANDT, PUNEX, SPECIE, STRNOR, THERMO, THERMV, THERM1, TOFENH, TOFH, TRANS
CPSV	1	PHASE1, PHYSOL, STRNOR
CRITER	20	CHEM, COEFF3, DRIVER, FREEMC, GAPPBI, INITP, MOCSOL, OUT, PHASE1, PHYSOL, PLUMIN, PPATPT, PRANDT, SOKFLX, STRNOR, TURN

Table 3-5 (Continued)

Common Block	Size (Decimal)	Routine Referencing
CROSS	2	STRNOR
CUTFO	6	DRIVER, FREEMC, ITARM, ITERM, PHASE1, PLMOUT, PLUMIN
DAT	800	INTERP, PARLOK, READF
DATAR	3602	ARASSL, AVERAG, AXIS, BACWRD, BLEXIT, BLMPIN, BOUND, BOUNDA, CBREAK, CHECK, CODEE, CODEF, CODEH, COEFEQ, COEFF3, DAMP, DECODE, DRIVER, EXPCOR, FIXIL, FNEWTN, FORWRD, FREEMC, FREPRO, GAPPBI, GASRD, GASTAP, HYPER, IBOUND, IMPUT, INITP, INTEGR, INTT, ITARM, ITERM, LAGRNG, LIMITS, LOGIC, MASS, MASSCK, MAXT, MCCRMK, MESH, MOCSOL, NORMCK, OFSET, ONED, OUT, OUTBIN, OUTPUT, OVEREX, PARINT, PARSTR, PARTIL, PARTPH, PHASE1, PHI, PHYSOL, PHYZOL, PLMOUT, PLOAD, PLUMIN, POINT, PPATPT, PRANDT, PRFRBD, PROP, PUNEX, REDIST, SLINT, SLSKIP, SOKFLX, SPECIE, STARTV, STGMOD, STLINE, STRNOR, THERMT, THRUST, TRACE, TRANS, WDGI, WG, WTFLOF, WTT, WXANDR
DBUG	4	HALL
DELT	1	DLTA, HALL, PARTIL
DFDR	13	BACKWRD, FORWRD, MCCRMK
DISCOM	2	DRIVER, INITP
DRAG	2	ONED, PARTIL, TRACE, TRANS

Table 3-5 (Continued)

Common Block	Size (Decimal)	Routine Referencing
DRAGCF	101	BLKDAT, DRAGCP, DRIVER, GAPPBI, PARTPH, PLMOUT, PLUMIN, PPATPT, TRACEP, TRANS
DROP	100	CHECK, PHASE1, PHYSOL, STRNOR
DRUG	512	BLKDAT, DRAGMR
EFHARY	29	BACWRD, CODEE, CODEF, CODEH, DECODE, FORWRD, MCCRMK,
ERR	1	PARTIL, TRANS
EVERY	2	PARLOK, READF
EXPER	5	HYPER, PAFH, PHASE1, POFH, PRANDT
EXPNN	1	GASRD, PLUMIN
FAB	3	DRIVER, FABLE, THERMO
FILIT	4	PARTIL
FIND	2	TRACEP
FLOW	2	PARLOK
FORCE	3	DRIVER, MASSCK, OUTBIN, THRUST
FREE	8	AVERAG, DRIVER, FREEMC, FREPRO, MOCSOL, PHASE1, PLMOUT, PLUMIN, PPATPT, STGMOD, STRNOR
FSTAG	806	AVERAG, BACWRD, CBREAK, CHECK, CODEE, CODEF, CODEH, DECODE, EXPCOR, FORWRD, FREEMC, FREPRO, GAPPBI, GASRD, HYPER, INTEGR, INTT, IRAD, LIPIN, MASSCK, MCCRMK, MOCSOL, OUT, OUTBIN, PARTIN, PHASE1, PHYSOL, PLMOUT, PLOAD, PPATPT, PRANDT, PRFRBD, SLINT, SLSKIP, STGMOD, STRNOR, THRUST

Table 3-5 · (Continued)

Common Block	Size (Decimal)	Routine Referencing
GAPPA	1442	AVERAG, BACWRD, BLEXIT, CHECK, CODEE, CODEF, CODEH, COEFEQ, COEFF3, DECODE, DRIVER, EXPCOR, FORWRD, FREPRO, GAPPBI, INTT, MCCRMK, MOCSOL, NEWENT, OUT, OUTBIN, PARINT, PARSTR, PBLTRC, PDT, PHASE1, PHYSOL, PLMOUT, PLUMIN, PPATPT, PRFRBD, PUNEX, SLINT, SPECIE, STGMOD, STRNOR, WRITP, WTFLOP
GASCON	5	AVERAG, BACWRD, BLEXIT, CODEE, CODEF, CODEH, COEFF3, DAMP, DECODE, DELTAF, DELTFF, DRIVER, EMOFP, EMOFV, ENTROP, ENTRPP, ESHOCC, ESHOCK, FABLE, FORWRD, FREEMC, FREPRO, GAPPBI, GASRD, HYPER, IMPUT, LIPIN, MCCRMK, MOCSOL, NORSCK, OUT, PATH, PARTIL, PARTIN, PHASE1, PHYSOL, PLMOUT, PLUMIN, POFLM, POFH, PPATPT, PRANDT, REDIST, RGMOFP, RGVOFM, RHOFEM, SETHTG, SITER, SOKFLX, STARTV, STGMOD, STRNOR, THERMO, THERMV,THERM1, THETPM, TOFEM, TOFENH, TOFH, TOFV, TRANS, VOFEM, WEAK, WEKK, WOFA
GASDAT	2562	BLEXIT, BLMPIN, CHEM, DRIVER, FABLE, GASRD, GASTAP, IDTAPE, IMPUT, LIPIN, OUT, OUTBIN, PARTIL, PARTIN, PARTPH, PHASE1, PLMOUT, PLUMIN, PUNEX, RGMOFP, RGMOFM, SETHTG, SLSKIP, SPECIB, SPECIE, TAB, THERMO, THERMT, THERMV, THERM1, TOB, TOFENH, TOFH, TRANS
GLOBAL	500	BACWRD, CHECK, CODEE, CODEF, CODEH, DECODE, EXPCOR, FORWRD, FREEMC, INTT, MCCRMK, MOCSOL, OUTBIN, PHASE1, PLOAD, STRNOR
GRINT	2	DRIVER, FABLE, THERMO
GSV	1	AVERAG, STGMOD

Table 3-5 (Continued)

Common Block	Size (Decimal)	Routine Referencing
HEAD	60	DRIVER, GASTAP, IDTAPE, INITP, OUT, PAGE, PAGVOF, PLMOUT, PLUMIN
HITWAL	11	BLEXIT, PBLTRC, WRITP, WTFLOP
HUL	1	GASRD, GASTAP, TRANS
IDL	1	BLEXIT, DRIVER, PLUMIN
ILINE	1	PARTIL
INPUT	800	LIPIN, PARTIN, PLMOUT, PLUMIN, STLINE
INTCR	10	GASRD, INTEGR, MASSCK, THRUST
INTEU	1	PHASE1, STRNOR
IPMX	1	BLEXIT, DRIVER, PBLTRC, WRITP
IRN	1	DECODE, MCCRMK, PHASE1
ISEA	1	PHASE1, RGMOFP, STRNOR
ISTRT	1	BLEXIT, DRIVER, OUTBIN, PLUMIN, PUNEX
ITOTI	1	PARLOK, READF
LIMIT	20	PARLOK, READF
LIPCOM	8	PARTIN, PLUMIN, SETHTG
LIPFX	1	PHASE1, STRNOR
LIPNT	2	STRNOR
LIPPT	12	BLMPIN, PHASE1
LSAD	11	CHECK, PARINT, PARSTR, PHASE1
LSTRLN	20	INTT, MCCRMK

Table 3-5 (Continued)

Common Block	Size (Decimal)	Routine Referencing
MASOUT	2	MASSCK, PHASE1
MASSC	11	DRIVER, MASSCK, PARTIN, PLMOUT, PLUMIN, TRANS
MET	2	PBLTRC, READF, SPACET, TAPMOV
MOL	32	AVERAG, DRIVER, FABLE, GASRD, PLMOUT, PLUMIN, THERMV
NAMEA	10	PARTIL, PROP, TRACE, TRANS
NAMEL	24	ONED, PARTIL, PROP, TRACE, TRANS, WDGI
NAMEM	7	ONED, PARTIL, TRACE, WDGI
NAMEQ	7	ONED, PARTIL, TRACE, TRANS
NAMER	43	FIXIL, ONED, PARTIL, PROP, TRACE, TRANS
NAMES	10	DLTA, FIXIL, ONED, PARTIL, PROP, TRACE, WDGI
NAMEW	2	PROP, TRANS
NAMEX	5	ONED, PARTIL, PROP, TRACE, TRANS
NAMEY	31	ONED, PARTIL, PROP, TRACE
NAME1	5	ONED, PARTIL, PROP, TRANS
NSF	10	DRIVER, MASSCK, PARTIN, PHASE1, STRNOR
OFSTAR	10	DRIVER, GASRD, GASTAP
ONTSPT	2	COEFEQ, COEFF3, DRIVER, MOCSOL, PARTIN, PPATPT, PRFRBD, STRNOR

Table 3-5 (Continued)

Common Block	Size (Decimal)	Routine Referencing
OVERLA	1	PHASE1, STRNOR
PARSTU	10	COEFF3, STRNOR
PARTP1	10000	MAIN
PARTP2	21	ARASSL, BACWRD, BLEXIT, BLMPIN, CHECK CODEE, CODEF, CODEH, DECODE, DRIVER, FABLE, FORWRD, FREEMC, GAPPBI, GASTAP, IDTAPE, INTEGR, INTT, MASSCK, MCCRMK, MOCSOL, NEWENT, OUT, OUTBIN, OVEREX, PARINT, PARLOK, PARSTR, PARTIL, PARTIN, PARTPH, PBLTRC, PFP, PHASE1, PHYSOL, PLMOUT, PLOAD, PLUMIN, PPATPT, PRANDT, PREAD, PRFRBD, PUNEX, SLINT, SLSKIP, SPECIB, SPECIE, STRNOR, THERMO, THERMV, THRUST, TRANS, WTFLOP
PARTP4	400	ARASSL, BACWRD, BLEXIT, CHECK, CODEE, CODEF, CODEH, COEFF3, DECODE, DRIVER, FORWRD, FREEMC, GAPPBI, INTEGR, INTT, MASSCK, MCCRMK, MOCSOL, OUT, OUTBIN, OVEREX, PARINT, PARSTR, PARTIN, PFP, PHASE1, PHYSOL, PLMOUT, PLOAD, PLUMIN, PPATPT, PRANDT, PREAD, PRFRBD, PUNEX, SLINT, SLSKIP, STRNOR, THERMO, THRUST
PARTTP	3	PLMOUT, PLUMIN
PARTWT	1	DRIVER, PLUMIN, PUNEX
PBLL	1	CODEH, DRIVER, PPATPT
PCTC	5	BLMPIN, GAPPBI, GASTAP, HYPER, IMPUT, LIPIN, MOCSOL, NORSCK, PAFH, PARTIL, PARTIN, PHASE1, POFH, PPATPT, PRANDT, SETHTG, THERM1, TOFENH, TOFH, TRANS

Table 3-5 (Continued)

Common Block	Size (Decimal)	Routine Referencing
PHISOL	40	PHYZOL, STRNOR
POINTC	2	CHECK, COEFF3, GAPPBI, MOCSOL, PHYSOL, STRNOR
PRAD	1623	IRAD, OUT, OUTBIN, PBLTRC, PHASE1, PLMOUT, PLUMIN, READF
PRNV	1	DRIVER
PRTINT	4	FREEMC
PSEC	4	COEFF3, PHASE1, STRNOR
PSLD	42	COEFF3, DRIVER, GAPPBI, MASSCK, PARTIN, PBLTRC, PLMOUT, PLUMIN, PPATPT, THRUST
PSTR	200	OUT, OUTBIN, PHASE1
PTEN	2	BLMPIN, PHASE1
PUNEXT	4	OUTBIN, PHASE1
PUT	1	CHECK, EXPCOR, PHASE1
PW1	1	PROP
QUITIT	1	PARLOK, PBLTRC
RSTART	2	DRIVER, PHASE1, PLUMIN
RUE	2	CHEM, STRNOR
RWTD	3	FIXIL, PROP, TRANS
RZMAP	3	FIXIL, PROP
RZW1	6	FIXIL, PARTIL, PROP, TRACE, TRANS, WDGI

Table 3-5 (Continued)

Common Block	Size (Decimal)	Routine Referencing
SAVE	10	GAS, TRACEP
SAVTEM	122	PARLOK
SIGMB	100	GASTAP, IMPUT, PLUMIN, SETHTG
SIGNAL	1	DRIVER, MASSCK, PLUMIN
SKIPPY	6	ARASSL, BLEXIT, BLMPIN, DRIVER, GASTAP, PHASE1, PLUMIN, PUNEX, SLSKIP
SLIPPT	40	COEFEQ, COEFF3, MOCSOL, NEWENT, PHYSOL, PHYZOL, PRFRBD, STRNOR
SLOW	1	TRACEP
SOURCE	2	PARTIL
SPECI	1650	ARASSL, PUNEX
SPEL	5000	ARASSL, BLEXIT, BLMPIN, DRIVER, IMPUT, LOGIC, MASS, OUTPUT, PARLOK, PARTIN, PBLTRC, PUNEX, SPCTX, SPECC, SPECIB, START, STARTV
SPFK	71	BLMPIN, SPECIB
STATN	61	INTERP, PARLOK
STEPC	10	CHECK, DRIVER, EXPCOR, FREEMC, INITP, MASSCK, MESH, MOCSOL, PHASE1, PLMOUT, PLUMIN, PRANDT, STRNOR, THETPM
STSUB	1	ADASTR, PLUMIN, STARTV
SVDT	3	GASTAP, IDTAPE, PLUMIN
TAPRIT	1	DRIVER, GASTAP, IDTAPE, IMPUT, OUTBIN, PARTPH, PLMOUT, PLUMIN

Table 3-5 (Continued)

Common Block	Size (Decimal)	Routine Referencing
TEMPER	1	AVERAG, DRIVER, EMOFP, FABLE, GAPPBI, OUT, PARTIN, PHYSOL, POFEM, PPATPT, STGMOD, THERMO, THERMV, THERM1, TOFH
TEMP01	24	BOUNDA, PHASE1, PHYZOL
TEMP02	40	PHASE1, PHYSOL, STRNOR
TEMP03	25	PHASE1
TFLAG	11	DRIVER, GAPPBI, OUTBIN, PARTPH, PBLTRC, PLUMIN, PPATPT, SLSKIP, TEMTAB
TOTAL	2	AVERAG, BACWRD, CBREAK, CODEE, CODEF, CODEH, COEFF3, DECODE, FORWRD, MCCRMK, OUT, PARINT, PARSTR, PHASE1, SLINT, STRNOR
TPEH	1040	DRIVER, PARTPH, PUNEX, TEMTAB, TRANS
TRCDAT	162	GAS, PARLOK, PBLTRC, START, WTFLOP
TRPRT	4	BLMPIN, DRIVER, EXPCOR, GASRD, GASTAP, IDTAPE, IMPUT, MASSCK, OUTBIN, PARTIN, PARTPH, PHASE1, PLUMIN, PUNEX, THRUST
TUIPA	146	STRNOR
UBVB	5	DLTA, FIXIL, HALL, PROP
VAROF	1	DRIVER, GAPPBI, GASRD
VARSL	1	DRIVER, IMPUT, PLUMIN
VISEX	3	GAPPBI, IMPUT, MOCSOL, NEWENT, PARTIN, PHASE1, PLUMIN, PPATPT, PUNEX, SETHTG, THERM1, TOFENH, TOFH, TRANS
VISSS	2	PUNEX, SPECIE

Table 3-5 (Concluded)

Common Block	Size (Decimal)	Routine Referencing
VLIM	161	DRIVER, GASRD, THERMO, THERMV
VMIX1	9	COEFEQ, NEWENT
VMIX2	50	CHEM, OUT
VMIX3	27	BLEXIT, CHECK, CHEM, COEFEQ, EXPCOR, MOCSOL, NEWENT, OUT, OUTBIN, PHASE1, PHYSOL, PLUMIN, PRANDT, PUNEX, STRNOR
VMIX4	3	CHECK, PHASE1
VMIX5	3	NEWENT
VMIX6	1	CHECK, PHASE1
VSON	2	DRIVER, GASTAP, TRANS
VTRY	16	FABLE, THERMV
WAFT	2	BOUND, DRIVER, LAGRNG, PHASE1, PLMOUT, PLUMIN
WALBD	7	BLMPIN
WALPR	2642	BLMPIN, WALPRP
WRITIT	1	PHASE1
WRITIT	1	DRIVER, OUT, PLUMIN
WT	1	DRIVER, MASSCK, PARTIN, PLMOUT, THRUST,
XSICOM	520	DRIVER, FABLE, GASRD, XSI
XXSH	1	COEFF3, DRIVER, GAPPBI, GASTAP, IDTAPE, PARTIL, PHASE1, PLUMIN, PPATPT

4. PROGRAM FILES

This section provides a list and description of all the files that are required to execute the entire RAMP2 programs. Table 4-1 presents a list of all the files along with their function, type of files (INPUT/OUTPUT), programs in which they are used and additional comments.

- DESCRIPTION OF THE UNFORMATTED BINARY OUTPUT OF THE RAMP2F FLOWFIELD TAPE (UNIT 3)

The binary tape output on Unit 3 is described in this section. Initial input data are written on the first part of the data tape and gaseous and particle data are written out for each data point in the flow field. This tape is formatted so that it may be used by other auxiliary routines (plot, plume impingement or radiation). When the RAMP2F program is sequentially executed for both the nozzle and plume the plume data are stored behind the nozzle data as if a single run has generated the flowfield.

GROUP I - General Information

Number of Records = 1

Write ( ) (HEADER(I), I=1,60),ISPECS,IMETRIC,RSTAR,LIN3,ICON1,NRSS

- HEADER
  - run identification (2A4)
  - date (3A4)
  - description (55A4)
- ISPECS = number of particle species to be considered
- IMETRIC* = 0 English flowfield units
  = 1 Metric flowfield units

*Determined from ICON(9)

Table 4-1 DESCRIPTION PROGRAM FILES USED BY RAMP2 CODES

Unit	Function	Input/Output	Program	Type	Comments
1	BLIMPJ Data	Output	RAMP2F	U*	This file is output by RAMP2F program and contains the input data to BLIMPJ program.
2	Boundary Layer Information	Output Input/Output Input	RAMP2F(1) BLIMPJ RAMP2F(2)	U U	This file contains the stations coordinates for the boundary layer and the results of the boundary layer solution for each boundary layer computational station. The RAMP2F(1) nozzle execution outputs this file. The BLIMPJ uses this file as input and outputs on this file. The RAMP2F(1) plume run uses this file as input.
3	Flowfield Data Scratch File	Output Input Input/Output Output/Input	RAMP2F(1) BLIMPJ RAMP2F(2) TRAN72	U U U U	This file is output by the RAMP2F(1) nozzle run and is used as input by the BLIMPJ code. The second RAMP2F(2) run uses this file as input then outputs the plume data behind the nozzle data (RAMP2F(1)).
4	TRAN72 Product Data	Input Input	TRAN72 RAMP2F(1)	F** F	This file contains the curve fit and transport data used by the TRAN72 program. The RAMP2(1) nozzle solution also uses this file to generate data for the BLIMPJ program.
5	Card Input	Input	TRAN72 RAMP2F	F F	Input data file for TRAN72 and RAMP2F(1) codes.
6	Printed Output	Output	TRAN72 RAMP2F BLIMPJ	F F F	Printed output file.
7	Punched Output	Output	TRAN72 RAMP2F	F F	Punched output file for TRAN72 and RAMP2F codes.
8	Startline Data Scratch File	Output/Input	RAMP2F(1),(2)	F	This file contains the startline generated by the RAMP2F code using either the single phase or two-phase transonic modules. This file could also contain a startline generated by another code if it is in the correct format.
10	Thermodynamic Data	Output Input	TRAN72 RAMP2F(1)	U U	TRAN72 cutfile of thermodynamic data used by RAMP2F(1) and RAMP2F(2) is input.
11	Scratch	Output/Input	RAMP2F(1) RAMP2F(2)	U U	Scratch File
12	Restart Startline and SPF Startline Data	Output Input	RAMP2F(1) RAMP2F(2)	U U	This is a file generated by RAMP2F(1) nozzle solution which contains the SPF exit plane and normal exit plane data which is used to generate the exit plane SPF and restart staartline which contains the boundary layer.
13	Input Data	Output Input	RAMP2F(1) RAMP2F(2)	F F	All input data (cards(tape 5)) for the RAMP2F(1) nozzle solution is output on this file and is read as input in place of File 5 for the RAMF2F(2) plume run.

*Unformatted; **Formatted.

- RSTAR = Throat radius (ft or meters)

- LIN3 = 1 single line output (See ICON(8) of Input)
  2 two line output
  3 three line output

- ICON1 = 1 gas properties read from cards
  2 gas properties read from tape
  3 finite rate chemistry
  4 same as 3 except frozen

- NRSS 0 not presently used.

GROUP II - Gas Data (Written only if ICON(1) < 3)

Number of Records = 1 + IOF*IS

```
Write ( )  (BETA(I), I=1, 6), IOF, IS
```

- BETA is gas identification name (6A4)

- IOF number of total enthalpy cuts through "Mollier chart" (max = 10)

- IS number of entropy cuts (max = 1)

```
DO M=1,IOF
DO I=1,IS

Write ( )  IV,IDATA, ((TEMP(J,K), K=1, IDATA), J=1, IV), IVT,
           ((CPM(J,K), K=1,5), J=1, IVT), RSTAR, PINF, EMINF,
           GAMINF, FINF, EXINF, XSHIFT, HTREF
```

- IV number of velocity cuts through "Mollier chart" for this total enthalpy and entropy + 2 (max = 15)

- IDATA number of gaseous species present for this total enthalpy and entropy (max = 98)

- GAMINF freestream isentropic exponent

- IVT = IV-2

- RSTAR throat radius (ft or m)

- PINF ambient pressure (psf or Newtons/m^2)

- EMINF freestream or external stream Mach number
- EXINF limit to which equation applies
- FINF linear static pressure gradient (slope) approach
- XSHIFT nozzle length (ft or m)
- HTREF total enthalpy (cal/gm) of gas table for 0.0 H_T table (see Vol. III, Section 4.)
- TEMP contains the following information for each value of IOF,IS

	1	2	3	4	5	6	7	8	9	. . . .	IDATA	
	P_f									. . . .		
1								(A4)		. . . .	(A4)	Species
2								(A4)		. . . .	(A4)	Name
3	Htg	P	T	S	ψ	γ	M_c	X			X_F	
4	|			|			M*	|		|	
5	|			|				|		|	
.	c			c					|		|	
.	o			o					|		|	
.	n			n					Mole		|	
.	s			s					Fractions		|	
.	t			t					|		|	
.	a			a					|		|	
.	n			n					|		|	
.	t			t					|		|	
.	|			|					|		|	
IV	Htg	P	T	S		M		X_1			X_F	

- Pf freeze pressure (atm)
- Htg total enthalpy of the gas (cal/gm)
- P pressure (atm)
- T temperature (K)
- S entropy (cal/gm-K)
- ψ molecular weight (gm/gm-mole)
- γ isentropic exponent
- M^* throat Mach number = 1
- M Mach number for this table entry

- CPM — contains the following information

	1	2	3	4	5
1	Pr	μ	C	Not	H
2				Used	
3					
.					
.					
.					
IVT					

- Pr — Prandtl number
- μ — viscosity (poise)
- C_p — specific heat at constant pressure (cal/gm-K)
- H — static enthalpy (cal/gm)

GROUP III - Finite Rate Chemistry (Only if ICON(1) $\geq$ 3)

Write ( ) NT,NS,PC,TC,PR,VISO,EX

- NT — number of temperature entries for thermodynamic data tables
- NS — number of chemical species
- PC — chamber pressure (atm)
- TC — chamber temperature (^{o}K)
- PR — Prandtl number
- VISO — chamber viscosity (lbf-sec/ft^2, kgm/m-sec)
- EX — viscosity exponent ($\mu = \mu_o(T/T_o)^{EX}$)

Write ( ) ((AID(I,LL),LL=1,2),I=1,NS)

- AID(I,1) — first half of species number (A4)
- AID(I,2) — second half of species name (A4)

Write ( ) NT, (TTB(L),L=1,NT), (CPTB(J,L),HTB(J,L),L=1,NT),J=1,NS)

- TTB — temperature for each task entry (^{o}K)

- CPTB specific heat for each table entry (cal/mole-^{o}K)
- GTB entropy for each table entry (cal/mole-^{o}K)
- HTB enthalpy for each table entry (cal/mole)

GROUP IV - Gas Particle Data

Number of Records = ISPECS+1

Write ( ) IDUM, ((PSP(I,J), I=1,2), J=1, ISPECS)

- IDUM dummy word
- PSP(1,J) mass density of j^{th} particle (slug/ft^3 or kgm/m^3)
- PSP(2,J) radius (ft or m)

DO I=1,ISPECS

Write ( ) NTAB1,TMELT,HSOL,HLIQ,(HFIT(N,1,I),HFIT(N,2,I),N=1,NTAB1)

- NTAB1 number of table entries for this species
- TMELT melt temperature (R or K)
- HSOL enthalpy before phase change (ft^2/sec^2-R or m^2/sec^2-K)
- HLIQ enthalpy after phase change (ft^2/sec^2-R or m^2/sec^2-K)
- HFIT(N,1,I) temperature (R or K)
- HFIT(N,2,I) enthalpy (ft^2/sec^2 or m^2/sec^2)

Note that if NTAB=1 species is ideal and HFIT(1,1,I) = C_{PL} (specific heat of liquid) and HFIT(1,2,I) = C_{PS} (specific heat of the solid).

GROUP V - Flowfield Data

Number of records = $1 + ILAST + \frac{ICON1}{3} * ILAST + \frac{LIN3}{3} * ILAST + ILIM$

Write ( ) (ILAST, I=1,7), ILIM,THRUST,AEXIT,IEXIT

- ILAST — number of data points on the following normal surface. If ILAST = 0 there is no information to follow
- ILIM — number of points that have particles present
- THRUST — thrust (lbf or Newtons)
- AEXIT — exit plane area (ft^2 or m^2)
- IEXIT — exit flag: 0 if not exit; 1 if exit

Write ( ) ((ITYPE,R,X,M,θ,S,μ,δ, Htg), I=1,ILAST), (V,I=1,ILAST) (W_g,I=1,ILAST),((ρ,P,T,γ,R), I=1,ILAST)

- ITYPE — identifies type of point (wall, shock, interior, etc.)

  0 input point
  1 interior point
  2 wall point
  3 free boundary
  4 upstream shock point
  5 Prandtl-Meyer point
  6 downstream shock point
  7 slip line
  8 shock intersection point
  9 vibrational mode frozen
  10 .otational mode frozen
  11 translational mode frozen

- R — radial coordinate (ft or m)
- X — axial coordinate (ft or m)
- M — Mach number
- θ — flow angle (rad)
- S — entropy (ft^2/sec^2-R or m^2/sec^2-K)
- μ — Mach angle (rad)
- δ — shock angle (rad)

- Htg — gas total enthalpy (ft^2/sec^2 or m^2/sec^2)
- V — velocity (ft/sec or m/sec)
- W_g — mass flow between this streamline and axis (slug/sec or kgm/sec)
- ρ — gas density ($slug/ft^3$ or kgm/m^3)
- P — pressure (lb_f-ft^2 or N/m^2)
- T — temperature (R or K)
- γ — isentropic exponent
- R — universal gas constant divided by molecular weight (ft^2/sec^2-R or m^2/sec^2-K)

DO I=1, ILAST

Write ( ) (SPECN,I=1,NS)

- NS — number of gas species — Output only for finite rate cases
- SPECN — species mole fractions

Write ( ) (PO,I=1,ILAST)

- PO — pitot total pressure — Output only if 3 line printout selected ICON(8)

DO I = ILIM*

Write ( ) ISP, ((U,V,T,H,), J=1,ISP),ILIMIT

- ISP — number of particle sizes at this point
- U — axial velocity component (ft/sec or m/sec)
- V — radial velocity component (ft/sec or m/sec)
- T — temperature (R or K)
- H — enthalpy (ft^2/sec^2 or m^2/sec^2)

*This record is written only for those points that have particles present.

- ρ particle density (slug/ft^3 or kgm/m^3)
- ILIMIT 0 not a limiting streamline
  1 is a limiting streamline

NOTE: The flowfield data are repetitively stored on tape as indicated above normal surface after normal surface. When ILAST = 0 the end of the data has been reached.

5. DESCRIPTION OF RAMP SUBROUTINES

This section describes the subroutines of the three programs which make up the RAMP2 program (RAMP2F, TRAN72, and BLIMPJ). Section 5.1 discusses the RAMP2F subroutines. Section 5.2 presents a description of the TRAN72 program routines and Section 5.3 gives a description of the BLIMPJ subroutines. Writeups for Sections 5.2 and 5.3 were taken from Refs. 3 and 4, respectively.

5.1 DESCRIPTION OF THE RAMP2F SUBROUTINES

This subsection contains a detailed description of each routine used in the RAMP2F program.

Described are:

- Function (if applicable) of each routine
- Calling sequence
- Common blocks and other routines used,
- The method used in performing the routine functions.

For your convenience, the routines are organized alphabetically.

NOTE: The following routines are not explained in this section as they comprise the two-phase transonic solution of the SPP code (Ref. 7) which is incorporated in the RAMP code. A complete description of each of these routines is contained in Ref. 7.

COEFS	ORTHS
DLTA	PARTIL
FIXIL	PROP
HALL	TRACE
ONED	WDGI

Additionally, the following routines are dummy routines whose references have been left in the program so that the functions these routines performed could at some future date be put back into the code.

BOUNDA	SLPLIN
CARCTR	SOKINT
IRAD	SOKSOL
REPEAT	VMODEL
VMOD1	VMOD2

FUNCTION NAME: **ALGINT**

DESCRIPTION

This routine performs a log interpolation between two values of a variable.

CALLING SEQUENCE

= ALGINT (H,R1,R2)

where H is the interpolation factor and R1 and R2 are the values of the variables between which the interpolation is being made.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON - None
UTILITY - None

METHOD OF SOLUTION

$A = \ell n (R1) + H * (\ell n (R2) - \ell n R1)$

$ALGINT = e^{A}$

FUNCTION NAME: **AOASTR**

DESCRIPTION

This function finds the Mach number corresponding to a given area ratio by one-dimensional theory. Real gas effects are considered in this calculation.

CALLING SEQUENCE

```
EM = AOASTR (OF,S,AOA,K1W1,K2W2)
```

where EM is the Mach number which exists, one-dimensionally, at an area ratio of AOA, an entropy S, and at an O/F ratio or total enthalpy, OF.

UTILITY ROUTINES AND COMMON REFERENCES

```
COMMON/STSUB/
ERRORS
ITSUB
RGVOFM
THERMO
WOFA
```

METHOD OF SOLUTION

The weight flow per unit area at Mach one is evaluated. An initial guess for the desired Mach number (depending on whether AOA is subsonic or supersonic area ratio) is made and ITSUB is initialized. An iterative solution of the equation FOFEM = AOA - WOFA1/WOFA(EM), driving FOFTEM to zero, is performed with the aid of ITSUB.

SUBROUTINE NAME: **ARASSL**

DESCRIPTION

This subroutine sorts a two-dimensional array containing a spatial description of exit plane flow properties from the axis (smallest radial location) to the nozzle lip (largest radial location).

CALLING SEQUENCE

```
CALL ARASSL (PFPARY)
```

where PFPARY is a dummy array not used in the subroutine.

UTILITY ROUTINES AND COMMON REFERENCES

```
COMMON/SKIPPY/          COMMON/DATAR/
COMMON/SPEL/            COMMON/PARTP2/
COMMON/PARTP4           COMMON/CONTRL/
COMMON/SPECI/
```

METHOD OF SOLUTION

The routine is entered with a spatial random distribution of points which specify the distribution of flow properties at the exit plane of a nozzle. The four arrays that contain the flow properties are:

```
TEMP(M,N)               IPFP(M,N,1)
PART(M,N,1)             SPEC(N,M)
```

At each M point in the exit plane (up to 100 points), the M position of each array specifies the specific flow variable (i.e., temperature, specific mole fraction, particle density, etc). The points are rearranged in the arrays such that the first M position of each array contains the flow properties at the axis and the remaining M locations of the array contain a monotonically increasing (with radial location) description of the flow properties.

SUBROUTINE NAME: **AVERAGE**

DESCRIPTION

This subroutine determines if the flow is translationally frozen based on Knudsen number for non-continuum flow and sets the appropriate gas total conditions.

CALLING SEQUENCE

CALL AVERAG(IS,J,N,K,ITYPE)

where IS is the base point streamline number on the J data surface, N is the streamline point on the K line for which the flow regime is to be determined and ITYPE is a flag which is returned to the calling routine to indicate the flow regime.

ITYPE	Flow Regime
1	Continuum
4	Translationally frozen

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/TOTAL/	THERMO
COMMON/GSV/	EMOFV
COMMON/GAPPA/	TOFV
COMMON/GASCON/	RHOFEM
COMMON/CONTRL/	STGMOD
COMMON/FREE/	TOFEM
COMMON/FSTAG/	POFEM
COMMON/DATAR/	
COMMON/MOL/	
COMMON/TEMPER/	
COMMON/CPMUK/	

METHOD OF SOLUTION

The average Knudsen number between the old streamline base point is calculated via the following equation:

$$Kn = .788539 \quad (\bar{M}^2/\bar{R}_E)\left|\ell nT_1 - \ell nT_2\right| /dS$$

where the properties are averaged between the old (1) and new (2) streamline points. The flow regime is determined by checking the calculated Knudsen number against the input Kundsen number for criteria translational freezing. Once the flow regime has been determined the appropriate specific heat ratio (gamma) and total conditions are calculated.

SUBROUTINE NAME: AXIS

DESCRIPTION

This subroutine is used by the single phase transonic module to calculate the flow properties at the nozzle axis.

CALLING SEQUENCE

CALL AXIS (I,J,K,L,PFPARY)

where

I	point number on the data surface
J	data surface number
K	old time step identifier (1 or 2)
L	new time step idetnifier (2 or 1)
PFPARY	array containing flow properties for the old and new timestep for all points in the transonic computational domain.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/
IDMPHI
PHI

METHOD OF SOLUTION

See Ref. 14.

SUBROUTINE NAME: **BACWRD**

DESCRIPTION

This subroutine performs the backward differencing step in the radial (vertical) direction in the McCormick two-step marching shock-capturing algorithm.

CALLING SEQUENCE

```
CALL BACWRD (JS,L,PFPARY,J)
```

where

JS	is the species index. JS = 1, refers to the gas phase and JS 1 refers to solid particles within a particular size range.
L	refers to grid lines in the marching (downstream) direction. L = JK refers to the last computed line, and L = KK refers to the line currently being computed.
PFPARY	is the array of particle properties for each point on the new (KK) and old (JK) line.
J	is the radial (vertical) grid point index.

UTILITY ROUTINES AND COMMON REFERENCES

```
COMMON/DFDR/
COMMON/EFHARY/
COMMON/DATAR/
COMMON/PARTP2/
COMMON/PARTP4/
COMMON/GAPPA/
COMMON/GLOBAL/
COMMON/TOTAL/
COMMON/GASCON/
COMMON/CHEMXX/
COMMON/FSTAG/
COMMON/CONTRL/
CODEE
CODEF
```

METHOD OF SOLUTION

Described in Section 5 of Volume I.

SUBROUTINE NAME: BLEXIT

DESCRIPTION:

The subroutine merges the results of the boundary layer solution at the exit plane with the inviscid nozzle results in order to generate a viscous exit plane startline for a plume solution.

CALLING SEQUENCE:

CALL BLEXIT (PFPARY)

where

PFPARY is an array containing particle properties.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/ISTRT/	RGVOFM
COMMON/CAPUR/	POFEM
COMMON/GASCON/	PREAD
COMMON/SKIPPY/	IDMPFP
COMMON/GASDAT/	PDT
COMMON/CONTRL/	PFP
COMMON/PARTP2/	ITSUB
COMMON/IPMX/	INRSCP
COMMON/SPEL/	TOFEM
COMMON/CHEMCN/	SITER
COMMON/DATAR/	IDMPDT
COMMON/PARTP4/	UOFEM
COMMON/BPRESW/	
COMMON/GAPPA/	
COMMON/HITWAL/	
COMMON/IDL/	
COMMON/VMIX3/	

METHOD OF SOLUTION

The routine merges the inviscid nozzle results and boundary layer solution to generate user selected exit plane start lines for restart at the exit plane or punches start lines for use in the Standard Plume Flowfield code (SPF) or Lockheed Method of Characteristics (MOC) code. The variables ISTART and IDLSTL are user inputs to determine kinds of startlines to be generated.

ISTART	Type of Startline
0	None
1	Sets up and punches a SPF startline
2	Sets up a normal startline and executes a plume solution
3	Sets up a normal startline and punches an SPF startline. Executes a plume restart

IDLSTL	Description
0	Will not generate a MOC startline.
1	Generates an ideal gas MOC start line. Prints start line.
2	Generates an ideal gas MOC startline and punches startline and mass flow averaged total conditions.

The selection of ISTART = 1 or 3 requires that ISKPY(Card 5) be greater than zero. In order to punch or print an MOC startline ISTART must be a 1 or 3.

The MOC startline that the program punches is directly usable by the MOC program only if some program modifications are made to account for the fact that the startline points include total enthalpy variations. Appendix B describes these changes.

This routine requires data which are stored on FORTRAN Unit 12 from the previous nozzle solution and UNIT 2 from the BLIMPJ execution.

The subsonic portion of the boundary layer is treated using the replacement layer method whereby this region is replaced with a constant Mach 1.01 zone that contains the same mass flow as the subsonic portion.

SUBROUTINE NAME: BLKDAT

DESCRIPTION:

This block data routine initializes the Kliegel (Ref. 9) and Crowe (Ref. 10) gas-particle drag coefficients which are used by the code.

CALLING SEQUENCE

None

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DRAGCF/
COMMON/DRUG/
UTILITY - None

METHOD OF SOLUTION

Not applicable

SUBROUTINE NAME: **BLMPIN**

DESCRIPTION

This routine sets up the input data that the BLIMPJ code require for generating the nozzle boundary layer. This routine is entered after the inviscid nozzle solution has been completed. The routine reads in variables specifying the type of boundary layer chemistry, wall boundary conditions, and combustion chamber species (if the thermodynamic data was read in from cards). The routine will also punch the BLIMPJ data if the user desires to execute the BLIMPJ code separately.

UTILITY ROUTINES AND COMMON REFERENCE

COMMON/CPMUK/	COMMON/SPFK/	COMMON/TRPRT/
COMMON/PCTC/	COMMON/PARTP2/	IDSPEC
COMMON/WALBD/	COMMON/WALPR/	IDMTOB
COMMON/BLMDAT/	COMMON/PTEN/	TOB
COMMON/CONTRL/	COMMON/DATAR/	SPECIB
COMMON/SKIPPY/	COMMON/SPEL/	WALPRP
COMMON/LIPPT/	COMMON/GASDAT/	

METHOD OF SOLUTION

Not applicable.

SUBROUTINE NAME: BOUND

DESCRIPTION

This subroutine finds the radial coordinate and flow angle (radians) for a given axial coordinate on an upper or lower solid boundary.

CALLING SEQUENCE

CALL BOUND (R,X,THETA,ITYPE,K1W1,K1W2)

where R is the radial coordinate, X is the known axial coordinate, THETA is the wall angle and ITYPE indicates whether upper or lower boundary equations are to be used.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/
COMMON/WAFT/
LAGRNG

METHOD OF SOLUTION

The common block region DATAR contains boundary equations or wall coordinates necessary to evaluate K and THETA. The two types of equations used are:

$r = a\,[\sqrt{b + cx + dx} + e]$ Conic Type 1

$r = ax^4 + bx^3 + cx^2 + dx + e$ Polynomial Type 2

When the upper or lower boundary is described by discrete points (R,X,THETA) subroutine LAGRNG is called to interpolate for the R and THETA of the point. The input fixed point variable ITYPE has a one or a two in the unit position which selects the upper (2) or lower (1) coefficients or points and control information. IEQNOW contains the number of the equation to be used.

SUBROUTINE: CBREAK

DESCRIPTION

This subroutine determines where the breakdown of continuum flow starts to occur.

CALLING SEQUENCE

CALL CBREAK (IFLG,J,K)

where IFLG = 0 if the breakdown criteria is never executed on a normal, IFLG = point number on the normal where the breakdown criteria exceeds .05, J is the previous data surface and K is the new data surface.

UTILITY ROUTINES AND COMMON BLOCKS

COMMON/CPMUK/
COMMON/DATAR/
COMMON/FSTAG/
COMMON/TOTAL/
COMMON/CONTRL/
THERMO

METHOD OF SOLUTION

Birds (Ref. 5) breakdown criteria

$$P = \frac{q}{\rho v} \frac{d\rho}{ds}$$

is calculated at each point on the normal. If the breakdown parameter of .05 is exceeded, the code points out the location on the normal where P = .05. The collision frequency v is calculated using the hard sphere relationship.

$$v = \frac{5}{4} \frac{\rho RT}{\mu}$$

SUBROUTINE NAME: CHECK

DESCRIPTION

This subroutine determines whether or not to add or delete streamline points based on user input mesh controls.

CALLING SEQUENCE

CALL CHECK (I,K,IS,J,IGO,ITOTK,ITOTJ,PFPARY)

where (I,K) and IS,J) are the two points against which the program is checking the mesh control constraints. IGO = -1 for checking deletion and greater than zero for inserting points. ITOTK and ITOTJ are the total number of points on the J and K normals. PFPARY is the array which contains the particle properties.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/POINTC/
COMMON/DATAR/
COMMON/GLOBAL/
COMMON/PARTP4/
COMMON/PARTP2/
COMMON/STEPC/
COMMON/CONTRL/
COMMON/GAPPA/
COMMON/FSTAG
COMMON/DROP/
COMMON/CHEMCN/
COMMON/VMIX4/
COMMON/VMIX6/
COMMON/LSAD/
COMMON/CHEMXX/
COMMON/VMIK3/
COMMON/SPCTX
COMMON/PFP
IDMPFP
GAPPBI
VMODEL

METHOD OF SOLUTION

See Section 3.5.1 (Volume III) for a description of mesh control parameters.

SUBROUTINE NAME: CHEM

DESCRIPTION

This routine evaluates the chemical reaction-rate equations to determine the new chemical species concentrations.

CALLING SEQUENCE

```
CALL CHEM (DXX,RHO,U,T)
```

where DXX is the distance along the gas streamline from the base point to the new point

RHO = gas density
U = gas velocity
T = gas temperature

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CRITER/	COMMON/CHEMYY/
COMMON/RUE/	COMMON/VXIX2/
COMMON/CONTRL/	COMMON/VXIX3/
COMMON/CHEMCN/	COMMON/BLMDAT/
COMMON/GASDAT/	TKEY
COMMON/CHEMXX/	SLDP

METHOD OF SOLUTION

The reaction rate equations for the various chemical reactions are solved simultaneously using an implicit finite differencing scheme (Ref. 6).

SUBROUTINE NAME: CODEE

DESCRIPTION

This subroutine computes the E vector of flow properties for use in the McCormick two-step marching shock-capturing algorithm. The E vector consists of the following flow properties:

$$E = \begin{matrix} \text{Gas Phase} \\ \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{matrix} \quad \begin{matrix} \text{Solid Phase} \\ \rho_i u_i \\ \rho_i u_i^2 \\ \rho_i u_i v_i \\ \rho_i v_i h_i \end{matrix}$$

where

ρ = gas density

u = gas axial (horizontal) velocity

v = gas radial (vertical) velocity

p = gas radial pressure

H = gas total enthalpy

ρ_i = solid particle density in terms of mass of particles per unit volume of solid/gas mixture for ith species.

u_i, v_i = velocity components of solid particles for ith species

h_i = enthalpy of solid particles for ith species.

CALLING SEQUENCE

```
CALL CODEE (JS,L,PFPARY,J,I)
```

where

JS is the species index. JS = 1 refers to the gas phase, and JS 1 refers to solid particles within a particular size range.

L refers to grid lines in the marching (downstream) direction. L = JK refers to the last computed line, and L = KK refers to the line currently being computed.

PFPARY is the array of particle properties generated in Subroutine PPATPT.

J is the radial (vertical) grid point index

I is an index used in identifying values used in obtaining finite differences. For example, I = 1 refers to values at the point J on the last computed line. I = 3 refers to values at J $\pm$ 1, depending on whether the differencing is forward or backward. I = 2 refers to values being computed at J on the next line by solving the finite difference equations.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/EFHARY/
COMMON/DATAR
COMMON/PARTP2
COMMON/PARTP4
COMMON/GAPPA
COMMON/GLOBAL
COMMON/TOTAL
COMMON/GASCON
COMMON/CHEMXX
COMMON/FSTAG
COMMON/CONTRL

METHOD OF SOLUTION

Described in Section 5 of Volume I.

SUBROUTINE NAME: CODEF

DESCRIPTION

This subroutine computes the F vector of flow properties for use in the McCormick two-step marching shock-capturing algorithm. The F vector consists of the following flow properties:

	Gas Phase	Solid Phase
	ρv	$\rho_i v_i$
$F =$	ρuv	$\rho_i u_i v_i$
	$\rho v^2 + p$	$\rho_i v_i^2$
	ρvH	$\rho_i v_i h_i$

where

ρ = gas density
u = gas axial (horizontal) velocity
v = gas radial (vertical) velocity
p = gas radial poressure
H = gas total enthalpy
ρ_i = solid particle density in terms of mass of particles per unit volume of solid/gas mixture for ith species.
u_i, v_i = velocity components of solid particles for ith species
h_i = enthalpy of solid particles for ith species.

CALLING SEQUENCE

```
CALL CODEF (JS,L,PFPARY,J,I)
```

where

JS is the species index. JS = 1 refers to the gas phase, and JS 1 refers to solid particles within a particular size range.

L refers to grid lines in the marching (downstream) direction. L = JK refers to the last computed line, and L = KK refers to the line currently being computed.

PFPARY is the array of particle properties generated in Subroutine PPATPT.

J is the radial (vertical) grid point index

I is an index used in identifying values used in obtaining finite differences. For example, I = 1 refers to values at the point J on the last computed line. I = 3 refers to values at J $\pm$ 1, depending on whether the differencing is forward or backward. I = 2 refers to values being computed at J on the next line by solving the finite difference equations.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DFDR/
COMMON/EFHARY/
COMMON/DATAR
COMMON/PARTP2
COMMON/PARTP4
COMMON/GAPPA
COMMON/GLOBAL
COMMON/TOTAL
COMMON/GASCON
COMMON/CHEMXX
COMMON/FSTAG
COMMON/CONTRL

METHOD OF SOLUTION

Described in Section 5 of Volume I.

SUBROUTINE NAME: CODEH

DESCRIPTION

This subroutine computes the H vector of flow properties for use in the McCormick two-step marching shock-capturing algorithm. The H vector consists of the following flow properties:

	Gas Phase	Solid Phase
	0	0
	$\sum_{i=1}^{ns} \rho_i A_i \Delta u_i$	$-\rho A_i \Delta u_i$
$H =$	$\sum_{i=1}^{ns} \rho_i A_i \Delta v_i$	$-\rho_i A_i \Delta v_i$
	$-\sum_{i=1}^{ns} \rho_i A_i (B_i - u \Delta u_i - v \Delta v_i)$	$\rho_i A_i (B_i + \Delta u_i^2 + \Delta v_i^2)$

where

ρ_i = solid particle density in terms of mass of solid particles per unit volume of solid/gas mixture for i^{th} species.

u_i, v_i = velocity components of solid particles for i^{th} species.

$\Delta u_i, \Delta v_i$ = $u_i - u$, $v_i - v$ = velocity difference between solid particles and gas for i^{th} species.

A_i = defined by Eq. (3.46), Volume I.

B_i = defined by Eq. (3.103), Volume I.

CALLING SEQUENCE

```
CALL CODEH (JS,L,PFPARY,J)
```

where

JS is the species index. JS = 1 refers to the gas phase, and JS 1 refers to solid particles within a particular size range.

L refers to grid lines in the marching (downstream) direction. L = JK refers to the last computed line, and L = KK refers to the line currently being computed.

PFPARY is the array of particle properties for the old (JK) and new (KK) data surface.

J is the radial (vertical) grid point index

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/EFHARY/
COMMON/DATAR
COMMON/PARTP2
COMMON/PARTP4
COMMON/GAPPA
COMMON/GLOBAL
COMMON/TOTAL
COMMON/GASCON
COMMON/CHEMXX
COMMON/FSTAG
COMMON/CONTRL
PPATPT

METHOD OF SOLUTION

Described in Section 5 of Volume I.

SUBROUTINE NAME: COEFEQ

DESCRIPTION

This subroutine calculates the coefficients CI and ClJ for use in the gas-particle system compatibility equation along the gas Mach lines. CI is the gas total enthalpy term and ClJ is the particle contribution to the equation.

CALLING SEQUENCE

CALL COEFEQ (M,IPA,IPB,IPC)

where M is equal to 1 for limiting streamlines, IPA is the base point number for the RRC,IPB is the base point number of the LRC and IPC is the new point number.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/SLIPPT/	COMMON/CHEMXX/
COMMON/DATAR/	COMMON/CHEMCN/
COMMON/GAPPA/	COMMON/VMIX1/
COMMON/ONTSPT/	COMMON/VMIX3
COMMON/AVPROP/	COMMON/AVPRP2/
COMMON/CONTRL/	UTILITY - None

METHOD OF SOLUTION

The following finite difference relations are used to solve for the coefficients

$$CI_{1,2} = \frac{\cos\bar{\alpha}_{1,2}}{\sin\bar{\alpha}_{1,2}\ \bar{q}^2_{1,2}}$$

and

$$ClJ_{1,2} = \left\{ \sum_{j=1}^{NP} \bar{\rho}^j_{1,2}\ \bar{A}^j_{1,2} \left[\pm (\bar{v}_{1,2} - \bar{v}^j_{1,2}) \cos\bar{\beta}_{1,2} \mp (\bar{u}_{1,2} - \bar{u}^j_{1,2}) \sin\beta_{1,2} + \frac{\bar{B}^j_{1,2}}{\bar{q}_{1,2} \sin\bar{\alpha}_{1,2}} \right] \right\} \frac{\Delta x_{1,2}}{\bar{\rho}_{1,2}\ \bar{q}^2_{1,2} \cos\bar{\beta}_{1,2}}$$

For a detailed description of the calculation procedure, see Volume I, Section 4.3.

SUBROUTINE NAME: COEFF3

DESCRIPTION

This subroutine calculates the new particle properties at the point under consideration, and the intersection of the particle streamlines through this point with the J-line.

CALLING SEQUENCE

```
CALL COEFF3(IC,IPRES(3),VERT,IS,JS,I,K,IH+1,JS,IH-1,JS,ITYPE3
IP1,IP2,IP3,P3,PG5,PFPARY)
```

where

IC - point identifier for intersection of particle streamlines with J-line (See Fig. 7-2b, Vol. I).

IPRES(3) - number of particle size groups present at new point.

VERT - 1.0 for interior or upper boundary point
- 0.0 for lower boundary point.

IS,JS - the streamline base point on the reference normal (J-Line).

I,K - The point being solved for on the new normal (K-line).

IH+1,JS - the first point above the (IS,JS) point on the reference normal (J-line).

IH-1,JS - the first point below the (IS,JS) point on the reference normal (J-line).

ITYPE3 - type of point which is being solved for. 11 - interior point, 21 - lower boundary point, 22 - upper boundary point.

IP1 - point identifier (5) of the intersection of the new point streamline with the J-line (see Fig. 4-2, Vol. I).

IP2 - point identifier (4) of the particle streamline intersection with the J-line (see Fig. 4-2, Vol. I).

IP3 - point identifier (3) of the new point on the K-line (see Fig. 4-2, Vol. I).

P3 - array containing location and flow properties of the new point (I,K).

PG5 - array containing location and flow properties of the streamline intersection on the J-line (IP1).

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/	COMMON/PSLD/
COMMON/DATAR/	COMMON/XXSH/
COMMON/TOTAL/	COMMON/GASCON/
COMMON/PARTP1/	COMMON/CPMUK/
COMMON/GAPPA/	COMMON/SLIPPT/
COMMON/ONTSPT/	IDMPFP
COMMON/POINTC/	PFP
COMMON/PARSTU/	INRSCT
COMMON/CRITER/	PPATPT
COMMON/PSEC/	GAPPB1

METHOD OF SOLUTION

For a detailed description of the calculation procedure, see Volume I, Section 7.1.

SUBROUTINE NAME: COEFS

DESCRIPTION

This subroutine computes the coefficients of the least-squares polynomial which best approximates a weighted set of data points.

CALLING SEQUENCE

CALL COEFS(J,C,ALPHA,BETA,KC,A,T1,T2,T3,IND2)

where:

J — is the number of low-order coefficients to be set equal to zero.

C — is the array of KC + 1 polynomial coefficients.

ALPHA — is the α array of KC elements.

BETA — is the β array of KC elements.

KC — is the degree of the polynomial for whih the coefficients are to be found. Also, KC + 1 is the number of elements in the arrays C, A, T1, T2, and T3. KC is also the number of elements in the ALPHA and BETA arrays.

A — is the computed coefficient array with KC + 1 elements.

T1 — is an array of KC + 1 elements used for temporary storage by COEFS. The contents upon return from COEFS are of no significance to the user.

T2 — is an array of KC + 1 elements used for temporary storage by COEFS. The contents upon return from COEFS are of no significance to the user.

IND2 — is an error indicator.
= -2, when J > KC.
= +2, when J $\leq$ KC.

UTILITY ROUTINES AND COMMON REFERENCES

None.

METHOD OF SOLUTION

See Section 5.6.12 of Ref. 7.

SUBROUTINE NAME: **DECODE**

DESCRIPTION

This subroutine decodes the E vector of flow field properties computed by the McCormick two-step marching shock-capturing algorithm.

CALLING SEQUENCE

```
CALL DECODE (JS,JK,KK,PFPARY,J)
```

where

JS is the species index. JS = 1 refers to the gas phase, and JS > 1 refers to solid particles within a particular size range.

JK is the line number for the last computed line.

KK is the line number for the line currently being computed.

PFPARY is the array of particle properties for the old (JK) and new (KK) data surface.

J is the radial (vertical) grid point index.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/EFHARY/	COMMON/CONTRL/
COMMON/DATAR/	COMMON/IRN/
COMMON/PARTP2/	COMMON/CHEQ/
COMMON/PARTP4/	COMMON/CPMUK/
COMMON/GAPPA/	THERMO
COMMON/GLOBAL/	EMOFV
COMMON/TOTAL/	TOFEM
COMMON/GASCON/	UOFEM
COMMON/CHEMXX/	IDMPFP
COMMON/FSTAG/	

METHOD OF SOLUTION

Described in Section 5 of Volume I.

FUNCTION NAME: **DELTAF, DELTFF***

DESCRIPTION

This function computes the turning angle through an oblique shock wave knowing the shock angle and the upstream Mach number.

CALLING SEQUENCE

```
DELTA = DELTAF (EPS,EM,K1W1,K1W2)
```

or

```
DELTA = DELTFF (EPS,EM,K1W1,K1W2)
```

where DELTA, the turning angle is found from the shock angle, EPS, and the upstream Mach number, EM. NOTE: The appropriate values of gas properties must be stored in common upon entry to this routine.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON/

METHOD OF SOLUTION

The oblique shock relationships are used to determine the turning angle through an oblique shock wave.

$$\delta = \varepsilon - \tan^{-1}\left\{\tan\varepsilon\left(\frac{1}{M^2\sin^2\varepsilon} + \frac{\gamma-1}{2}\right)\left(\frac{2}{\gamma+1}\right)\right\}$$

*DELTAF and DELTFF are the same routine but are included as two separate routines in the program because of overlay requirements and the need to keep the core size as small as possible.

SUBROUTINE NAME: DLTA

DESCRIPTION

This subroutine is part of the two-phase transonic module. It modifies the gas phase transonic results along the supersonic startline so that the mass flow ratio, C_D, across this line is the same as given by Levine (Ref. 8) for the nozzle throat plane.

CALLING SEQUENCE

Call DLTA(RW1,ZW1Z,ZAX,RRT,VK)

where

RW1 — the radial coordinate (non-dimensionalized by the throat radius) of the starting line at the nozzle wall.

ZW1Z — the axial coordinate (non-dimensionalized by the throat radius) of the starting line at the nozzle wall.

ZAX — the axial coordinate (non-dimensionalized by the throat radius) of the starting line on the axis of the nozzle.

RRT — the radius of curvature (non-dimensionalized by the throat radius) of the nozzle upstream of the throat.

VK

$$= \left(1 + \frac{\dot{w}_p}{\dot{w}_g}\ \frac{Cp_\ell}{Cp_g}\right)/\left(1 + \frac{\dot{w}_p}{\dot{w}_g} * \frac{Cp_\ell}{Cp_g \gamma}\right)$$

γ = specific heat ratio
$\dot{w}_p$ = mass flow rate of particle
w_g = mass flow rate of gas
Cp_ℓ = specific heat of particles
Cp_g = specific heat of gas

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DELT/
COMMON/NAMES/
COMMON/UBVB/
PROP

METHOD OF SOLUTION

A value for C_D is obtained by numerically integrating across the initial line, then the flow speed is adjusted so that the integrated value matches the following throat value:

$$C_D = 1 - \frac{\gamma + 1}{(1 + R_c)^2} \quad \frac{1}{96} \quad \frac{8\gamma - 27}{2304(1 + R_c)} + \frac{754\gamma^2 - 757\gamma + 3633}{276480(1 + R_c)^2}$$

FUNCTION NAME: DOTPRD

DESCRIPTION

This function calculates the dot product of two vectors and returns the result to the calling routine.

CALLING SEQUENCE

= DOTPRD(V1,V2)

where V1 and V2 are any two vectors.

UTILITY ROUTINES AND COMMON REFERENCES

None.

METHOD OF SOLUTION

Vector V1 is dotted into vector V2. The resultant is a scalar returned as DOTPRD.

FUNCTION NAME: **DRAGCP**

DESCRIPTION

This routine determines the drag coefficient F ($C_D/C_{D_{Stokes}}$) as a function of Reynolds number.

CALLING SEQUENCE

= DRAGCP (RE)

where

RE is the particle Reynolds number.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DRAGCF/
UTILITY - None.

METHOD OF SOLUTION

$C_D/C_{D_{Stokes}}$ is tabulated as a function of particle Reynolds number and a linear interpolation is performed based on Reynolds number to obtain $C_D/C_{D_{Stokes}}$. This tabulation is that of Kliegel (Ref. 9).

FUNCTION NAME: DRAGMR

DESCRIPTION

This subroutine determines the local drag coefficient ($C_D/C_{D\,Stokes}$) as a function of particle Reynolds number and particle Mach number.

CALLING SEQUENCE

= DRAGMR (EM,RE)

where

EM is the particle Mach number
RE is the particle Reynolds number.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DRUG/
ALGINT

METHOD OF SOLUTION

$C_D/C_{D\,Stokes}$ as presented by Crowe (Ref.10) is tabulated as a function of particle Reynolds number and Mach number. A logarithmic interpolation is performed based on RE and EM to obtain the appropriate value of $C_D/C_{D\,Stokes}$.

SUBROUTINE NAME: **DRIVER**

DESCRIPTION

DRIVER provides the highest order control for program execution. The initialization and logic subroutines are called from here. Most of the common storage needed in the remainder of the program is specified here.

CALLING SEQUENCE

CALL DRIVER (K,K1W1,K1W2)
where K is a control constant indicating whether or not errors exist in the execution of the program. (K = 1 for a detected error, K = 0 for no errors.) K1W1 and K1W2 are flags which have various uses in the code.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/ACOM/	COMMON/IPMX/	COMMON/TEMPER/
COMMON/AUX/	COMMON/ISTRT/	COMMON/TFLAG/
COMMON/CAPUR	COMMON/MASSC/	COMMON/TPEH/
COMMON/CHEQ/	COMMON/MOL/	COMMON/TRPRT/
COMMON/CONTRL/	COMMON/NSF/	COMMON/VAROF/
COMMON/CPMUK/	COMMON/OFSTAR/	COMMON/VARSL/
COMMON/CRITER/	COMMON/ONTSPT/	COMMON/VLIM/
COMMON/CUTPD/	COMMON/PARTWT/	COMMON/VSON/
COMMON/DATAR/	COMMON/PARTP2/	COMMON/WAFT/
COMMON/DISCOM/	COMMON/PARTP4/	COMMON/WRITPT/
COMMON/DRAGCF/	COMMON/PBLL/	BLEXIT
COMMON/FAB/	COMMON/PRNV/	BLMPIN
COMMON/FORCE/	COMMON/WT/	INITP
COMMON/FREE/	COMMON/PSLD/	PBLTRC
COMMON/GAPPA/	COMMON/RSTART/	PHASE1
COMMON/GASCON/	COMMON/SIGNAL/	PLUMIN
COMMON/GASDAT/	COMMON/SKIPPY/	PUNEX
COMMON/GRINT	COMMON/SPEL/	SPACET
COMMON/HEAD/	COMMON/STEPC/	TRANS
COMMON/IDL/	COMMON/TAPRIT/	

METHOD OF SOLUTION

Not applicable.

SUBROUTINE NAME: **DUMSYS**

DESCRIPTION

This subroutine zeros out the particle array (PFPARY).

CALLING SEQUENCE

CALL DUMSYS (PFPARY)

where PFPARY is the array which will be used to store particle properties.

UTILITY ROUTINES AND COMMON REFERENCES

None.

METHOD OF SOLUTION

Not applicable.

FUNCTION NAME: **EMOFP**

DESCRIPTION

This routine computes the local Mach number as a function of local pressure (static) and local entropy.

CALLING SEQUENCE

EM = EMOFP (P,S,K1W1,K1W2)

where EM is the resultant Mach number found from the pressure, P, and entropy, S. NOTE: The appropriate values of the gas properties must be stored in COMMON upon entry to this routine.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/TEMPER/
COMMON/GASCON/
UTILITY - None

METHOD OF SOLUTION

Thermally perfect gas relationships are used to find the Mach number.

$$M = \sqrt{\left[\left(\frac{P_o\, e^{-S/R}(T_o/T_c)^{\gamma/\gamma-1}}{P}\right)^{\gamma-1/\gamma} - 1\right]^{\frac{2}{\gamma-1}}}$$

FUNCTION NAME: **EMOFV**

DESCRIPTION

This routine finds Mach number as a function of local velocity.

CALLING SEQUENCE

EM = EMOFV (V,K1W1,K1W2)

where EM is the local Mach number found as a function of the local velocity, V. NOTE: The appropriate values of the gas properties must be stored in COMMON upon entry to this routine.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON/
TOFV

METHOD OF SOLUTION

Thermally perfect gas relationships are used to find the Mach number.

$$M = \sqrt{\left(\frac{T_o}{T} - 1\right)\left(\frac{2}{\gamma - 1}\right)}$$

FUNCTION NAME: ENTROP, ENTRPP*

DESCRIPTION

This routine utilizes the oblique shock relations to find the entropy rise across a shock as a function of the shock angle and the upstream Mach number.

CALLING SEQUENCE

```
SD = ENTROP (EPS,EMU,K1W1,K1W2)
SD = ENTRPP (EPS,EMU,K1W1,K1W2)
```

where SD is the entropy rise across the shock and is a function of the shock angle, EPS, and the upstream Mach number, EMU. NOTE: The appropriate values of the gas properties must be stored in COMMON upon entry to this routine.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON/
UTILITY - None

METHOD OF SOLUTION

The oblique shock relations are employed to find the entropy rise across the shock.

$$ds = \frac{R}{\gamma-1} \ln\left[\frac{(2\gamma M^2 \sin^2\varepsilon - (\gamma-1))}{\gamma+1}\right] + \gamma \ln\left[\frac{\tan(\varepsilon-\delta)}{\tan\varepsilon}\right]$$

*ENTROP and ENTRPP are the same routine but are included as two separate routines in the program because of overlay requirements and the need to keep the core requirements as small as possible.

C-2

SUBROUTINE NAME: ERRORS

DESCRIPTION

ERRORS contains print messages for various errors which may occur. This is an open ended routine in that it can easily be extended to handle more print messages.

CALLING SEQUENCE

CALL ERRORS (I,K1W1,K1W2)

where I selects the message to be printed.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/
UTILITY - None.

METHOD OF SOLUTION

Not applicable.

SUBROUTINE NAME: ESHOCK,ESHOCC*

DESCRIPTION

This subroutine employs an iterative solution to perform the equilibrium shock calculations for a real or ideal gas. The real and ideal gas calculations are similar, the difference being that an ideal gas case converges on the first iteration.

CALLING SEQUENCE

```
CALL ESHOCK (OF,S1,V1,EP,DELTA,S2,V2,KZW,K1W)
CALL ESHOCC (OF,S1,V1,EP,DELTA,S2,V2,KZW,K1W)
```

where the input properties are, OF, the upstream O/F ratio or total enthalpy, S1,V1, the upstream entropy and velocity and, EP, the shock angle. The subroutine returns with DELTA, the turning angle and S2,V2, the downstream entropy and velocity.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/	POFEM
COMMON/GASCON/	DELTAF
EMOFV	ENTROP,ENTRPP
THERMO	RHOFEM
WEAK,WEKK	

METHOD OF SOLUTION

The continuity equation coupled with the equations for conservation of normal and tangential momentum are solved in an iterative manner utilizing thermochemical property data to satisfy the conservation of energy equation. This set of four equations is expressed in terms of the four unknown quantities:

ε = shock angle

δ = turning angle

S_2 = entropy downstream of shock

V_2 = velocity downstream of shock

*ESHOCK and ESHOCC are the same routines but are included as two separate routines in the program because of overlay requirements and the need to keep the core requirements as small as possible.

SUBROUTINE NAME: EXPCOR

DESCRIPTION

EXPCOR calculates the flow properties of those field points near an expansion corner.

CALLING SEQUENCE

CALL EXPCOR (NPM,J,K,ITOTJ,ITOTK,IPNT,K2W,K1W,PFPARY)

where

NPM = number of Prandtl-Meyer expansion rays emanating from the expansion corner

J = known normal line upstream of the expansion corner

K = the normal line under consideration downstream of the expansion corner

ITOTJ = adjusted total number of points on the J-line, not including the Prandtl-Meyer expansion points NPM

ITOTK = number of points on K-line before the Prandtl-Meyer expansion points are added; returns to the calling routine with the total number of points on K-line including Prandtl-Meyer expansion points

IPNT = indicates if an upper (=2) or lower (=1) boundary is being considered.

PFPARY = array containing particle properties.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CHEMXX/	COMMON/CAPUR/
COMMON/GLOBAL/	COMMON/FSTAG/
COMMON/CONTRL/	COMMON/GAPPA/
COMMON/DATAR/	INRSCT
COMMON/INPUT/	MOCSOL
COMMON/STEPC/	SPCTX
COMMON/AUX/	PPATPT
COMMON/VMIX3/	CHECK
COMMON/TRPRT/	VMODEL
COMMON/PUT/	ALGINT

METHOD OF SOLUTION

Flow properties at the expansion corner points are known (from PRANDT). Calculation starts from one of the corner points which have zero turning angle and proceeds toward the point with an increasing turning angle. Subroutine MOCSOL is used to solve for the flow properties of the intersection of the characteristic lines from two known points. The properties of the intersection of the normal from the known point on the new line (K-line, normal to the streamlines), with the characteristic of the corresponding point at the corner, are then interpolated. This point is then used along with another point at the expansion corner to find another new point, and so forth. The last of the expansion corner points is used twice in the calculation to find two points on the new normal - one on the characteristic line, the other on the streamline.

For a detailed description of the calculation procedure, see Volume I, Section 7.7.

SUBROUTINE NAME: FABLE

DESCRIPTION

This subroutine utilizes real or ideal gas information obtained from a master tape or input cards to calculate properties locally in the flow. The maximum size of the array used by FABLE is limited to eight gas properties ($V, R, \gamma, T_0, P_0, \mu, Pr, C_p$) at 13 velocity "cuts" for each of two entropy cuts and 10 O/F or total enthalpy cuts.

CALLING SEQUENCE

CALL TABLE (SS,VV,IF)

where SS is the local entropy, IF is the O/F or enthalpy table of interest and VV is the local velocity at the point of interest.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/XSICOM/	COMMON/GASDAT/
COMMON/CONTRL/	COMMON/MOL/
COMMON/GASCON/	COMMON/VTRY/
COMMON/FAB/	TOFV
COMMON/GRINT/	POFEM
COMMON/TEMPER/	EMOFV
COMMON/CPMUK/	XSI
COMMON/PART2/	TAB

METHOD OF SOLUTION

The routine is entered with an O/F or enthalpy table, IF, the local entropy, SS, and velocity, VV. A test is then made to determine if the gas is real or ideal. If the test indicates an ideal gas, the local properties are set to those stored in the TABB common array. If the test indicates real gas, a double interpolation scheme is utilized to locate gas properties between tabulated values of velocity and entropy. In the case of an entry beyond the range of the tables, an ideal gas extrapolation from the last table value is made to determine the gas properties.

SUBROUTINE NAME: FIXIL

DESCRIPTION

This subroutine locates the wall and centerline points for the supersonic startline based on user input Mach numbers.

CALLING SEQUENCE

CALL FIXIL

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/
COMMON/NAMER/
COMMON/NAMES/
COMMON/RWTD/
COMMON/RZMAP/
COMMON/RZW1/
COMMON/UBVB/
PROP

METHOD OF SOLUTION

Uses transonic results to establish where the startline intersects the nozzle wall and centerline based on user input values of Mach number.

FUNCTION NAME: FNEWTN

DESCRIPTION

This function solves for the Newtonian impact pressure along the plume boundary. The calculation is applicable for all freestream velocities including quiescent conditions (i.e., $M_\infty = 0$).

CALLING SEQUENCE

P_{IM} = FNEWTN (THETA3,X,ITYPE1,K1W1,K1W2)

where P_{IM} is the hypersonic Newtonian impact pressure at the plume boundary, THETA3 is the local flow angle at the boundary, X is the axial coordinate of the boundary point, and ITYPE indicates if an upper (=2) or lower (=1) boundary is being considered.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/
UTILITY - None

METHOD OF SOLUTION

The common block region WALLCO contains the necessary information to evaluate the freestream gas properties at the plume boundary point. The impact pressure is then calculated using the following equation

$$P = P_\infty (1 + eX) \left[1 + \gamma_\infty M_\infty^2 \sin^2(\theta_B - \theta_\infty)\right]$$

SUBROUTINE NAME: FORWRD

DESCRIPTION

This subroutine performs the forward differencing step in the radial (vertical) direction in the McCormick two-step marching shock-capturing algorithm.

CALLING SEQUENCE

```
CALL FORWRD (JS,L,PFPARY,J)
```

where

JS	is the species index. JS = 1, refers to the gas phase and JS > 1 refers to solid particles within a particular size range.
L	refers to grid lines in the marching (downstream) direction. L = JK refers to the last computed line, and L = KK refers to the line currently being computed.
PFPARY	is the array of particle properties on the old (JK) and new (KK) data surface.
J	is the radial (vertical) grid point index.

UTILITY ROUTINES AND COMMON REFERENCES

```
COMMON/DFDR/
COMMON/EFHARY/
COMMON/DATAR/
COMMON/PARTP2/
COMMON/PARTP4/
COMMON/GAPPA/
COMMON/GLOBAL/
COMMON/TOTAL/
COMMON/GASCON/
COMMON/CHEMXX/
COMMON/FSTAG/
COMMON/CONTRL/
CODEE
CODEF
```

METHOD OF SOLUTION

Described in Section 5 of Volume I.

SUBROUTINE NAME: **FREEMC**

DESCRIPTION

This subroutine computes flowfield properties in the free molecular regime.

CALLING SEQUENCE

CALL FREEMC (I1,J1,K1,ITOT,IOO,IOUT,IMOD,PFPARY)

where I1 is the point number for the first free molecular point on a normal, J1 is the old data surface, K1 is the new data surface, ITOT is the total number of points on the line, IOO is the line number for which a complete line is to be printed, IOUT is the total number of lines to skip between complete printout and IMOD is the number of points to shift on the old data surface to locate each base point streamline. PFPARY is the array which contains particle properties.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/	COMMON/GLOBAL/
COMMON/DATAR/	COMMON/PRTINT/
COMMON/PARTP2/	BOUND
COMMON/PARTP4/	ITARM
COMMON/FREE/	INRSCT
COMMON/GASCON/	WTFLOF
COMMON/FSTAG/	IDMPFP
COMMON/CUTFO/	PFP
COMMON/STEPC/	OUT
COMMON/CRITER/	OUTBIN

METHOD OF SOLUTION

Once it has been determined that a point is free molecular all successive calculations of the particular streamline point are made via FREEMC. The point properties are determined assuming that temperature, gas velocity, flow angle, gas constant and specific heat ratio (γ) are constant along a streamline. The gas density is determined from a source flow calculation (i.e., conservation of mass between streamlines).

$$\rho_2 = \frac{\rho_1 \, u_1 \, A_1}{u_2 \, A_2}$$

where subscript 1 is the old data surface properties and subscript 2 is the new data surface properties. The pressure at the new point is then determined from the equation of state.

SUBROUTINE NAME: **GAPPBI**

DESCRIPTION

This subroutine interpolates for the gas and particle properties between two known data points.

CALLING SEQUENCE

CALL GAPPBI (I8,JU,I9,KU,JB,M,ISKIPG,PG,FACTOR,M1,PFRARY)

where I8 is the base point number, JU is the base point line number, I9 is the seccnd point number, KU is the second point line number, JP is the temporary location in the IPFP array to store the interpolated data, M is the number of particles present, ISKIPG is a flag used to determine what arrays to use to do the interpolation, PG is the array in which the interpolated point properties are stored, FACTOR is the interpolation factor, and M1 = 0 gas only, M1 = 1 particles present. PFPARY is the array which contains particle properties.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/TFLAG/
COMMON/DATAR/
COMMON/PARTP4/
COMMON/PARTP2/
COMMON/GAPPA/
COMMON/GASCON/
COMMON/POINTC/
COMMON/CPMUK/
COMMON/CONTRL/
COMMON/FSTAG/
COMMON/TEMPER/
COMMON/CRITER/
COMMON/PSLD/
COMMON/XXSH/
COMMON/CAPUR/
COMMON/VAROF/
COMMON/DRAGCF/
COMMON/PCTC/
COMMON/VISEX/
ALGINT
PFP
THERMO
UOFV
TOFV
EMOFV
POFEM
TEMTAB
DRAGMR
DRAGCP

METHOD OF SOLUTION

The routine performs a linear interpolation between the properties of two known points and stores the results in temporary arrays which are used in other parts of the program during the calculation. ISKIPG is a flag which tells GAPPBI which arrays to use for the interpolation and whether or not to interpolate on particle properties.

SUBROUTINE NAME: **GAS**

DESCRIPTION

This subroutine looks up gas properties from the BLIMPJ boundary layer results. The gas properties which are determined are used by the routines which trace particle properties through the boundary layer.

CALLING SEQUENCE

```
CALL GAS (X,Y,PFPARY)
```

where X and Y are the coordinates in the nozzle where boundary layer properties are to be determined. PFPARY is the array which contains the spatial variation of boundary layer properties in the nozzle.

UTILITY ROUTINES AND COMMON REFERENCES

```
COMMON/TRCDAT/
COMMON/CHOCK/
COMMON/SAVE/
PRO
```

METHOD OF SOLUTION

A linear interpolation is performed between stations based on the axial location (X) and the non-dimensional boundary layer thickness $(R_N-R)/(R_N - R_{BL})$ where R_N is the nozzle wall radial coordinate at X and R_{BL} is the location of the boundary layer edge at X.

SUBROUTINE NAME: **GASRD**

DESCRIPTION

This subroutine reads in the gas properties. These properties may be real or ideal and read in via cards or tape. The routine also converts input gas properties from MKS units to English (ENG) units if necessary.

CALLING SEQUENCE

```
CALL GASRD (IPAR)
```

where IPAR is a 1 for two-phase flow and a zero for gas-only flow.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/XSICOM/	COMMON/HUL/
COMMON/CONTRL/	COMMON/TRPRT/
COMMON/INTCR/	COMMON/VLIM/
COMMON/GASCON/	COMMON/EXPNN/
COMMON/GASDAT/	COMMON/OFSTAR/
COMMON/MOL/	COMMON/CHEQ/
COMMON/FSTAG/	ITAPE
COMMON/DATAR/	IDMTB
COMMON/VAROF/	TAB

METHOD OF SOLUTION

The gas name, ALPHA(I), type units, number of O/F tables and number of entropy cuts are read in from an input card. If the gas properties are on cards, this subroutine reads the cards. If the gas properties are on tape, control of the reading of properties is given to GASTAP. In either case, the properties are converted from MKS to English (ENG) units by this subroutine if necessary.

SUBROUTINE NAME: GASTAP

DESCRIPTION

GASTAP reads the real gas properties from the thermochemical data tape generated by the modified TRAN72 computer program and writes these same data on a flowfield tape for communication with other programs.

CALLING SEQUENCE

CALL GASTAP

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASDAT/	COMMON/TAPRIT/
COMMON/CONTRL/	COMMON/SIGMB/
COMMON/DATAR/	COMMON/VSON/
COMMON/HEAD/	COMMON/SKIPPY/
COMMON/PARTP2/	COMMON/HUL/
COMMON/CHEMCN/	COMMON/OFSTAR/
COMMON/PCTC/	IDMTAB
COMMON/XXSH/	ERRORS
COMMON/BPRESW/	IMPUT

METHOD OF SOLUTION

The gas name, ALPHA(I), specified on the input data is compared with available cases on the TRAN72 thermochemical data tape until a match is found. This particular case is then read, stored in core, arranged in a form such that automatic transmission of data to other programs is possible, and then written on the RAMP flowfield tape.

SUBROUTINE NAME: HALL

DESCRIPTION

The subroutine establishes gas flow properties at any given point in the transonic region using the method of Hall (Ref. 11) as corrected and modified by Kliegel and Levine (Ref. 8).

CALLING SEQUENCE

CALL HALL (IORDER,RR,ZZ,RRT,VK,XMACH,THETA,ZERO)

where

IORDER	the order of approximation, 1, 2 or 3. 3 is default.
RR	radial position in transonic region
ZZ	axial portion in transonic region
RRT	upstream wall radius of curvature
VK	q, specific heat ratio
XMACH	Mach number at RR,ZZ
THETA	flow angle at RR,ZZ
ZERO	0 for first entry 1 for subsequent entries

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DBUG/
COMMON/DELT/
COMMON/DBVB/

DESCRIPTION

A bivariant interpolation scheme is used to match the two-dimensional results with one dimensional conditions at the nozzle inlet. See Section 5.6.33 of Ref. 7 for more detail.

SUBROUTINE NAME: HYPER

DESCRIPTION

This subroutine calculates the balanced pressure at a corner point (i.e., at the intersection of a solid boundary and the pressure boundary). The pressure balance is determined for either the overexpanded or underexpanded case with impact or ambient freestream pressure.

CALLING SEQUENCE

CALL HYPER (PB,I,K,ITYPE1,K1W1,K1W2,PFPARY)

where PB is the boundary pressure. I,K locates the boundary point, and ITYPE1 indicates if an upper (=2) or lower (=1) boundary is being considered. PFPARY is the array which contains particle properties.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON/	EMOFV
COMMON/CONTRL/	FNEWTN
COMMON/DATAR/	OVEREX
COMMON/FCTC/	ITSUB
COMMON/FSTAG/	THETPM
COMMON/EXPER/	TOFH
THERMO	PAFH
POFEM	ERRORS

METHOD OF SOLUTION

The boundary pressure (may be impact or ambient) is compared with the static pressure at the corner point. Depending on whether the comparison indicates the flow is overexpanded or underexpanded, a branch is made to OVEREX or THETPM. In either of these routines an iterative process balances the boundary pressure with the flowfield pressure at the boundary.

FUNCTION NAME: IBOUND

DESCRIPTION

The function determines which boundary equation the program should be using to solve the boundary point.

CALLING SEQUENCE

IEQNOW(J) = IBOUND(X,J)

where X is the axial coordinate and J is a 1 for a lower boundary and a 2 for an upper boundary.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/
COMMON/DATAR/

METHOD OF SOLUTION

Not applicable.

FUNCTION NAME: IDMPDT

DESCRIPTION

This function computes the storage location in the PDAT array of the particle properties in the boundary layer.

CALLING SEQUENCE

= IDMPDT(I,J,K,L)

where I,J,K,L are indices used to determine the storage location.

UTILITY ROUTIES AND COMMON REFERENCES

None.

METHOD OF SOLUTION

IDMPDT = I + 10 * (J-1 + 7 * (K-1 + 10 * (L-I))).

FUNCTION NAME: IDMPFP

DESCRIPTION

This function computes the particle storage location within the PFPARY array.

CALLING SEQUENCE

= IDMPFP (I,J,K,L)

where I,J,K,L are indices used to determine the storage location.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/PARTP1/
COMMON/PARTP2/
COMMON/PARTP3/
RWU

METHOD OF SOLUTION

The particle storage location is computed using the following relation

IDMPFP = I + 5 * (J-1 + 10 * (K-1 + 100 * (L-1))

FUNCTION NAME: IDMPHI

DESCRIPTION

This function calculates the storage location in the PFPARY array of the location and flow properties of the grid points in the single phase transonic solution.

CALLING SEQUENCE

= IDMPHI (I,J,K,L)

where I,J,K,L are indices used to determine the storage location.

UTILITY ROUTINES AND COMMON REFERENCES

None.

METHOD OF SOLUTION

IDMPHI = I + 12 * (J-1 + 15 * (K-1 + 27 * (L-1))).

FUNCTION NAME: IDMPOP

DESCRIPTION

This function calculates the storage location in the PFPARY array of the specie mole fractions for the equilibrium or equilibrium/frozen thermodynamic data which were generated by the TRAN72 for use in the RAMP solution.

CALLING SEQUENCE

= IDMPOP(I,J,K,L)

where I, J,K,L are indices used to determine the storage location.

UTILITY ROUTINES AND COMMON REFERENCES

None.

METHOD OF SOLUTION

IDMPHI = I + 10 + (J-1 + 2 * (K-1 + 25 * (L-1)))

FUNCTION NAME: IDMPRO

DESCRIPTION

This function calculates the storage location in the PFPARY array of the boundary layer properties which are used to trace particles through the BLIMPJ boundary layer results.

CALLING SEQUENCE

= IDMPRO(I,J,K)

where I,J,K are indices used to determine the storage location.

UTILITY ROUTINES AND COMMON REFERENCES

None.

METHOD OF SOLUTION

IDMPRO = I + 14 * (J-1 + 15 * (K-1))

FUNCTION NAME: **IDMTAB**

DESCRIPTION

This function computes the gas property storage location within the TABB array.

CALLING SEQUENCE

= IDMTAB (I,J,K,L)

where I,J,K,L are indices used to determine the storage location.

UTILITY ROUTINES AND COMMON REFERENCES

None.

METHOD OF SOLUTION

The gas property storage location, is computed using the following relation

IDMTAB = I + 10 * (J-1 + 2 * (K-1 + 13 * (L-1)))

FUNCTION NAME: **IDMXSI**

DESCRIPTION

This function computes the gas interpolacion parameter storage location within the XSIDIM array.

CALLING SEQUENCE

= IDMXSI (I,J,K,L)

where I,J,K,L are indices used to determine the storage location.

UTILITY ROUTINES AND COMMON REFERENCES

None.

METHOD OF SOLUTION

The gas interpolation parameter storage location is computed using the following relation

IDMXSI = I + 10 * (J-1 + 2 * (K-1 + 13 * (L-1))).

SUBROUTINE NAME: IDTAPE

DESCRIPTION

This subroutine writes the gas properties which were input via cards on the flowfield program tape. The format used to write them on tape is compatible with that used for a real gas.

CALLING SEQUENCE

CALL IDTAPE (UNITS, K1W1,K1W2)

where UNITS indicates whether the gas properties are being read in with English (ENG) or MKS units.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/XXSH/	COMMON/BPRESW/
COMMON/TAPRIT/	COMMON/PARTP2/
COMMON/CONTRL/	COMMON/GASDAT/
COMMON/HEAD/	COMMON/TRPRT/
COMMON/SVDT/	TAB

METHOD OF SOLUTION

When gas properties are read in from cards the data are converted to MKS units and then written on the flowfield tape (Unit 3).

FUNCTION NAME: **IDSPEC**

DESCRIPTION

This function calculates the storage location of the species mole fractions for the equilibrium or equilibrium/frozen thermodynamic results which are used to determine the species mole fraction which will be used in the BLIMPJ boundary layer solution.

CALLING SEQUENCE

```
= IDSPEC (I,J,K,L)
```

where I,J,K,L are indices used to determine the storage location.

UTILITY ROUTINES AND COMMON REFERENCE

None.

METHOD OF SOLUTION

```
IDSPEC = I + 2 * (J-1 + 2 * (K-1 + 13 * (L-))).
```

SUBROUTINE NAME: **IMPUT**

DESCRIPTION

This routine reads the input cards or tape for the chemistry package. The reaction rate equations, rate constants, and startline species concentrations are read in or determined from the TRAN72 data tape and the appropriate conversions, if any, are performed. Tables of enthalpy, entropy, and specific heats for each species are also input.

CALLING SEQUENCE

CALL IMPUT (IDATA)

where IDATA specifies the proper index of the array being input from a CEC data tape from which species concentrations are being extracted.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/
COMMON/CHEMCN/
COMMON/DATAR/
COMMON/GASDAT/
COMMON/CHEMXX/
COMMON/CPMUK/
COMMON/VISEX/
COMMON/PCTC/
COMMON/GASCON/
COMMON/VARSL/
COMMON/TAPRIT/
COMMON/SPEL/
COMMON/TRPRT/
COMMON/SIGMB/
SPECTX

METHOD OF SOLUTION

The routine reads species thermodynamic data and constructs a Gibbs free energy array to replace the entropy array. The reaction rate constant data, reactions, and third body data are input and stored. Finally, the startline species concentrations are input via cards or tape and converted to mole/mass ratios.

SUBROUTINE NAME: INITP

DESCRIPTION

This subroutine initializes the values of various control parameters, thereby providing for proper operation of the program. These initial values include:

1. The counter for the upper and lower boundary equations,
2. The counter for the first characteristic line,
3. The initial number of degrees per Prandl-Meyer ray,
4. Convergence criteria, and
5. Maximum number of iterations.

CALLING SEQUENCE

CALL INITP (K1W1,K1W2)

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTROL/
COMMON/CRITER/
COMMON/DATAR/
COMMON/DISCOM/
COMMON/HEAD/
COMMON/STEPC/
UTILITY - None

METHOD OF SOLUTION

Not applicable.

SUBROUTINE NAME: **INRSCT, INRSCP***

DESCRIPTION

INRSCT finds the intersection of two straight lines.

CALLING SEQUENCE

```
CALL INRSCT (T1,T2,T3,T4,T5,T6,R3,X3,K1W1,K1W2),
```

or

```
CALL INRSCP (T1,T2,T3,T4,T5,T6,R3,X3,K1W1,K1W2)
```

where T1,T2,T3 and T4,T5,T6 define the equations of the two straight lines which intersect at R3,X3.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON - None
ERRORS

METHOD OF SOLUTION

The equations of the straight lines are written

$$r = \tan T3\ (x - T2) + T1$$

and

$$x = \cot T6\ (r - T4) + T5$$

These equations are solved for x, but a test on the slopes is made to prevent indeterminate forms. If an indeterminate form is possible, the points are mapped one onto another, thus precluding the possibility of indeterminancy except when the lines are parallel.

*INRSCT and INRSCP are the same routine but are included as two separate routines in the program because of overlay requirements and the need to keep the core requirements as small as possible.

SUBROUTINE NAME: INTEGR

DESCRIPTION

This suboroutine calculates the incremental force and energy between two adjacent points in the flow field.

CALLING SEQUENCE

```
CALL INTEGR (DELX,DELY,THTBR,R,DA,V,
RHO,P,X,I,K,FXP,FYP,TRP,FXG,FYG,TRG,
AXO2D,ENU,EG,EP,EM,DW,PFPARY)
```

where

DELX	=	difference in axial position between the two points
DELY	=	difference in radial position between the two points
THTBR	=	averge flow angle of the two points
R	=	average radial position of the two points
DA	=	absolute distance between the two points
V	=	average gas velocity of the two points
RHO	=	average gas density of the two points
P	=	average gas pressure of the two points
X	=	average axial position of the two points
I	=	point number of the base point
K	=	line number of the base point
FXP	=	incremental force in axial direction due to the particle momentum
FYP	=	incremental force in radial direction due to the particle momentum
TRP	=	incremental torque due to particle momentum
FXG	=	incremental force in axial direction due to gas
FYG	=	incremental force in radial direction due to gas
TRG	=	incremental torque due to gas axial and radial forces
AXO2D	=	geometric term for axisymmetric or 2-D flow
ENU	=	angle the line connecting the two points has referenced to horizontal
EG	=	incremental gas energy
EP	=	incremental particle energy
EM	=	sum of incremental particle and gas energy (i.e., mixture)
DW	=	incremental gas mass flow between the two points
PFRARY	=	array containing particle properties

UTILITY ROUTINES AND COMMON REFERENCE

COMMON/CONTRL/
COMMON/PARTP4/
COMMON/PARTP2/
COMMON/DATAR/
COMMON/FSTAG/
COMMON/INICR/
PFP
VEMAG

METHOD OF SOLUTION

This subroutine calculates the mass flow, energy, momentum, and thrust produced by the particles and gas contained in each streamtube bounded by two streamline points on a normal. The resulting values are integrated along each normal and compared to the initial data surface to determine how well the solution is conserving the conservation equations.

SUBROUTINE NAME: INTERP

DESCRIPTION

This routine performs a linear interpolation for particle properties at a given axial location in the nozzle.

CALLING SEQUENCE

CALL INTERP (I,J,K,L,IPOINT,X,PFPARY)

where (I,K) and (L,J) are the streamline points on two consecutive normals that bracket the axial position (X) at which particle properties are required. IPOINT is a dummy variable that is no longer used. PFPARY is the array which contains the particle properties.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DAT/
COMMON/STATN/

METHOD OF SOLUTION

The particle properties and location on streamline at a specified axial station (X) are determined using a linear interpolation using the streamline points (I,K) and (L,J) which bracket X. The particle properties are determined from the PFRARY array. This routine is used to help determine the particle properties at the boundary layer edge after the nozzle solution and boundary layer solution have been performed.

FUNCTION NAME: **ITERM, ITARM***

DESCRIPTION

ITERM tests each normal lower wall point to determine if it is within the predefined problem limits. If the point falls outside the limits, the case is terminated.

CALLING SEQUENCE

```
FUNCTION = ITERM (IP,K,K1W1,K1W2)
FUNCTION = ITARM (IP,K,K1W1,K1W2)
```

where IP identifies the characteristic point on the new K line.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CUTFO/
COMMON/DATAR/
UTILITY - None

METHOD OF SOLUTION

The angular orientation of a line drawn from the upper or lower cutoff coordinates to the characteristic point is determined. One can determine if the point is inside or outside the problem limits by comparing this angle to the angle of the upper or lower cutoff line.

*ITERM and ITARM are the same routines but are included as two separate routines in the program because of overlay requirements and the need to keep the core requirements as small as possible.

SUBROUTINE NAME: **ITSUB**

DESCRIPTION

This subroutine controls the iterative solution of any set of equations which can ultimately be expressed as a function of one variable; it can also be used to control an integration loop.

CALLING SEQUENCE

```
CALL ITSUB (FOFY,Y,SAVE,CONV,NTIMES,K1W1,K1W2)
```

where

FOFY is the function of Y which is driven to zero
Y is the variable which is iteratively solved for
SAVE is the program control array, i.e., SAVE(1) is a control counter,
SAVE(2) is the Y increment
CONV is the convergence criteria for FOFY
NTIMES = maximum number of iterations to be performed

UTILITY ROUTINES AND COMMON REFERENCES

None.

METHOD OF SOLUTION

ITSUB modifies Y in the proper direction by the increment value SAVE(2) until the root has been bracketed. The method of false position is then used to modify Y until the solution is reached. Immediately after entering ITSUB each time, the function is inspected for convergence. If the function has converged, a program control is set, and computer control is transferred to the calling routine.

SUBROUTINE NAME: **KIKOFF**

DESCRIPTION

This subroutine terminates the use if an error in the calculation is encountered.

CALLING SEQUENCE

```
CALL KIKOFF (K1W1,K2W2)
```

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/
UTILITY - None

METHOD OF SOLUTION

Not applicable.

SUBROUTINE NAME: LAGRNG

DESCRIPTION

This subroutine determines the radial location and flow angle for solid boundaries which are input as tables of R,X, and flow angle.

CALLING SEQUENCE

```
CALL LAGRNG (IER,ID,ARG,R,THETA,ITYPE)
```

where

IER is an error flag, ID is a table location,
ARG is the axial value for which the radial coordinate, R, of the wall and flow angle, THETA, at the wall are desired,
ITYPE indicates if an upper (=2) or lower (=1) boundary is being considered.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/
COMMON/CONTRL/
COMMON/WAFT/
UTILITY - None

METHOD OF SOLUTION

The routine uses the Lagrange interpolation formula to solve for R and flow angle as a function of axial position, X, from a set of tabular points describing a solid boundary. The routine uses the two closest points to the desired X to solve the interpolation formula. In the vicinity of large nonlinear variations in R and flow angle the points should be placed close together.

SUBROUTINE NAME: **LIMITS**

DESCRIPTION

This subroutine tests the new boundary point to determine if it is within the limits of the current boundary equation. Depending on the test, the options are:

1. Use the current boundary equation,
2. Advance to the next boundary equation, or
3. The current equation is the last one specified.

CALLING SEQUENCE

```
CALL LIMITS (I,K,ITYPE,IOK,K1W1,K1W2)
```

where I,K represents the location of the boundary point in the PHO array, ITYPE indicates if an upper or lower boundary is being considered, and IOK is a control indicating if option 1, 2 or 3 is to be used.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/
COMMON/DATAR/
BOUND

METHOD OF SOLUTION

The radius, RMAX, and boundary angle, THETAMAX, at the limiting axial value XMAX is calculated in BOUND. RMAX or XMAX is compared with R or X for the point in question. The results of the comparison determine which of options 1, 2, or 3 is to be used.

SUBROUTINE NAME: LIPIN

DESCRIPTION

LIPIN calculates information for the starting line points when the simplified straight start line option is used (i.e., when ICON(2)≠2).

CALLING SEQUENCE

```
CALL LIPIN (COOR,S,INTOT,DELM,K1W1,K1W2)
```

where COOR is the starting line information array, S is the entropy level of the start line; INTOT is the total number of input points specified (50 max), DELM is Mach number gradient along the startline, and K1W1 is a flag which determines the type of startline point distribution.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/INPUT/
COMMON/CONTRL/
COMMON/PCTC/
COMMON/FSTAG/
COMMON/GASCON/
COMMON/GASDAT/
RGVOFM
UOFV
THERMO

METHOD OF SOLUTION

The startline input data are divided into the specified number of increments. Radial gradients in Mach number, X and θ, are calculated.

K1W1 = 0 The startline points are evenly spaced
K1W1 = 1 The startline points are concentrated near the upper boundary.

PROGRAM NAME: **MAIN**

DESCRIPTION

This is the main program which initiates the overall program solution.

CALLING SEQUENCE

None.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/PARTP1/
DUMSYS
DRIVER

METHOD OF SOLUTION

Not applicable.

SUBROUTINE NAME: **MASCON**

DESCRIPTION

MASCON calculates the Mach number distribution at an area downstream of the throat such that total mass flow is conserved. Mass flow, calculated at the throat, is used as the constant for comparison.

CALLING SEQUENCE

```
CALL MASCON (E,SE,DELM,K1W1,K1W2)
```

where E is the input line array CORLIP, SE is the input line entropy level, and DELM is the Mach number gradient along the startline.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/
RGVOFM
ERRORS
EMOFV
ITSUB
RHOFEM

METHOD OF SOLUTION

The mass flow rate the throat, m^*, is calculated. This m^* is compared with that at the input line location for an initial Mach number distribution. The Mach number distribution is then perturbed until mass flow is conserved.

SUBROUTINE NAME: **MASS**

DESCRIPTION

The routine is used by the single phase transonic module to calculate the mass flow and momentum at each solution station.

CALLING SEQUENCE

CALL MASS (WDOT,PDOT,JN,KN,PFPARY)

where WDOT and PDOT are the mass flow and momentum at the JN station at the new timestep KN. PFPARY is the array containing all the flow properties at the old and new time step for all points in the transonic flow domain.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/
COMMON/SPEL/
PHI
THERMT

METHOD OF SOLUTION

The mass and momentum fluxes are integrated starting at the wall point and proceeding to the axis at each station at a given time.

SUBROUTINE NAME: MASSCK

DESCRIPTION

This subroutine keeps a running check on the mass flow. Mass flow at the starting line is calculated and compared with that crossing each normal line downstream.

CALLING SEQUENCE

CALL MASSCK (ILAST,ISTART,K,K1W1,K1W2,PFPARY)

where ILAST is the last point on the normal line, ISTART is a number of the first point on the normal and K represents the normal line under consideration. PFPARY is the array containing the particle properties.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/	COMMON/FORCE/
COMMON/MASSC/	COMMON/WT/
COMMON/PSLD/	COMMON/PARTP4/
COMMON/CONTRL/	COMMON/PART2/
COMMON/NSF/	COMMON/INTCR/
COMMON/STEPC/	COMMON/FSTAG/
COMMON/SIGNAL/	INTEGR
COMMON/MASOUT/	PFP
COMMON/TRPRT/	IDMPFP

METHOD OF SOLUTION

The mass flow through the startline is calculated and stored. Mass flow through lines downstream is calculated and these values compared with the initial value. A percent change in mass flow is printed for each normal line. The total mass flow passing under each point on a characteristic line is stored so the mass flow can be written on the output tape to permit streamline tracing.

SUBROUTINE NAME: **MAXT**

DESCRIPTION

This routine finds the smallest time step that must be taken to maintain stability during the single phase transonic solution.

CALLING SEQUENCE

CAKK MAXT(DTCK)

where DTCK is the smallest time step which is found in the array which contains the maximum stable time step at all points in the solution domain.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/

METHOD OF SOLUTION

The smallest time step over all the solution points is located and returned to the calling routine.

SUBROUTINE NAME: MCCRMK

DESCRIPTION

This subroutine performs the McCormick two-step marching shock-capturing algorithm. In performing this algorithm, calls are made to subroutines CODEE, CODEF and CODEH to code the E, F and H vectors of flowfield properties. The forward and backward finite differences in the radial (vertical) direction and performed by calls to subroutines FORWRD and BACWRD. And finally, the flowfield properties as computed by the finite difference equations are obtained by decoding the "next-line" E vector by making a call to subroutine DECODE.

CALLING SEQUENCE

CALL MCCRMK (JK,KK,PFPARY)

where

JK is the line number for the last computed line.

KK is the line number for the line currently being computed.

PFPARY is the array of particle properties for the new (KK) and old (JK) data surface.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/EFHARY
COMMON/DATAR/
COMMON/PARTP2/
COMMON/PARTP4/
COMMON/GAPPA/
COMMON/GLOBAL/
COMMON/TOTAL/
COMMON/GASCON/
COMMON/CHEMXX/
COMMON/FSTAG/
COMMON/CONTRL/
COMMON/IRN/
CODEE
CODEH
FORWRD
BACWRD
DECODE

METHOD OF SOLUTION

Described in Section 5 of Volume I.

SUBROUTINE NAME: MESH

DESCRIPTION

This routine calculates the maximum initial step size that can be taken following a restart at the exit plane using a startline which contains a boundary layer.

CALLING SEQUENCE

```
IPNFD = MESH(L)
```

where L is the total number of points on the start line and IPNFD is the point number on the startline (starting from the axis) where the Mach number is less than 2.0.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/
COMMON/STEPC/
THERMO
EMOFV

METHOD OF SOLUTION

This routine finds the first point away from the nozzle lip where the Mach number is less than and closest to Mach 2.0. The maximum step size is then taken to be the distance from this point to the lip. A Mach number of 2 was chosen so that a reasonable initial step size can be taken while maintaining numerical accuracy.

SUBROUTINE NAME: MOCSOL

DESCRIPTION

This subroutine solves the characteristic equations for gas only flow in the region around and downstream of an expansion corner.

CALLING SEQUENCE

CALL MOCSOL (IN,KN,IN1,KN1,IN2,KN2,IFLAG,ITYPE,K1W1,K1W2,PFPARY)

where IN,KN identifies the storage location for the new point to be computed, IN1,KN1 identifies the right running known point, and IN2,KN2 identifies the left running known point. IFLAG is an error indicator and ITYPE selects the type calculation. PFPARY is the array containing the particle properties.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/PARTP4/	COMMON/STEPC/	COMMON/CHEMX4/	BOUND
COMMON/PARTP2/	COMMON/CPMUK/	COMMON/VXIX3/	ROTERM
COMMON/GAPPA/	COMMON/PCTC/	COEFF3	VOFEM
COMMON/ONTSPT/	COMMON/CONTRL/	INRSCT	RGMOFP
COMMON/AVPROP/	COMMON/CRITER/	POFEM	FNEWTN
COMMON/SLIPPT/	COMMON/DATAR/	COEFEQ	NEWENT
COMMON/GLOBAL/	COMMON/GASCON/	PPATPT	ERRORS
COMMON/FSTAG/	COMMON/POINTC/	PFP	SPCTX
COMMON/FREE/	COMMON/VISEX/	IDMPFP	VMODEL
			ALGINT
			THERM1

METHOD OF SOLUTION

The four characteristic equations are written as a function of five variables, R,X,θ, V, and S. An additional relationship is obtained by assuming the entropy, S, varies linearly between known data points. Using these characteristic equations in finite difference form, the routine solves for a new mesh point, knowing two mesh points of an opposite family.

The solution is begun by setting the average values of properties over the step length equal to the known values at the base point. Subsequent passes in the iterative solution result in "updated" average values. The iterative solution is continued until the desired convergence on velocity or flow angle is reached or until the maximum number of iterations is exceeded.

MOSCOL is utilized by subroutine EXPCOR to solve the normal line immediately downstream of any expansion corner.

SUBROUTINE NAME: NEWENT

DESCRIPTION

This subroutine calculates the change in entropy and gas total enthalpy along a gas streamline for gas particle flows.

CALLING SEQUENCE

```
CALL NEWENT (NP,IT1,IT2,S3,H3,K,PB)
```

where

NP = number of particles present on streamline
IT1 = 1 for interior point
= 2 for wall point
IT2 = 1 for interior or lower wall point
= 2 for upper wall point
S3 = entropy at new point
H3 = total enthalpy or O/F ratio at new point
K = 5 gas only streamline
= 7 gas and particles present on streamline
PB = array containing streamline base point properties (upstream)

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL	COMMON/CHEMCN/
COMMON/CHEMXY/	COMMON/VISEX/
COMMON/AVPROP/	COMMON/VMIX1/
COMMON/GAPPA/	COMMON/VMIX3/
COMMON/SLIPPT/	COMMON/VMIX5/
COMMON/CHEMXX/	COMMON/PARTP2/
	CHEM

METHOD OF SOLUTION

The compatibility relations for gas total enthalpy and entropy (Eqs. (4.2) and (4.3) of Table 4-1, Vol. I) are solved at the new streamline point knowing the gas and particle properties at the new and base streamline points. For gas only flows (and streamlines not crossing a shock) the gas total enthalpy and entropy are held constant along a given streamline.

FUNCTION NAME: **NORMCK**

DESCRIPTION

This routine checks to see whether the start line is close enough to a true normal to proceed with the solution assuming the startline is a normal.

CALLING SEQUENCE

INM = NORMCK(L)

where L is the number of points on the start line. NORMCK = 0, the line is close to a normal. NORMCK=1, the line is not close to a normal.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/

METHOD OF SOLUTION

A new normal is constructed using the streamline angles from the startline and an assumed step size equal to the axial distance between the first and last startline points. If the difference in the axial step size between the upper wall points exceeds 1-1/2 times the centerline axial step then the solution will not begin at the axis until the region above the start line is calculated.

SUBROUTINE NAME: NORSCK

DESCRIPTION

This routine uses local flow properties to calculate properties downstream of a normal shock to obtain pitot pressure. This routine is used only for finite rate chemistry real gas cases.

CALLING SEQUENCE

```
CALL NORSCK (VI,PI,EMI,TI,GMI,RI,HI,POSTR)
```

where

VI,PI,.....,HI are the local values of velocity, pressure, Mach number, temperature, gamma, gas constant and enthalpy

POSTR is the pitot pressure.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON/
COMMON/PCTC/
TOFENH
ITSUB

METHOD OF SOLUTION

The downstream conditions are first estimated using ideal gas relations. The routine then performs an iteration as follows:

1. Calculate downstream static enthalpy from energy equation.
2. Iterate in subroutine TOFENH for temperature, gamma and gas constant.
3. Calculate downstream pressure from continuity and equation of state.
4. Check to see if resultant pressure satisfies the Rayleigh line equation. If not, increment the downstream velocity and repeat steps 1 through 4.
5. When the iteration is complete, the pitot pressure is determined from the downstream conditions.

FUNCTION NAME: **OFSET**

DESCRIPTION

The function determines the oxidizer-to-fuel ratio on the initial data surface for the single-phase transonic module.

CALLING SEQUENCE

```
OF = OFSET(R)
```

where R is the radial coordinate of the point on the initial data surface and OF is the OF ratio at the point.

UTILITY ROUTINES AND COMMON REFERENCES

```
COMMON/DATAR/
```

METHOD OF SOLUTION

A linear interpolation is performed using the local radial coordinate and an input table of O/F vs radial position.

SUBROUTINE NAME: ONED

DESCRIPTION

This subroutine integrates the relations for one-dimensional two-phase perfect gas nozzle flow through the nozzle conical inlet assuming the flow is directed toward a sink point at the apex of the cone.

CALLING SEQUENCE

CALL ONED

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DRAG/	COMMON/NAMER/
COMMON/DATAR/	COMMON/NAMES/
COMMON/NAMEL/	COMMON/NAMEX/
COMMON/NAMEM/	COMMON/NAMEY/
COMMON/NAMEQ/	COMMON/NAME1/

METHOD OF SOLUTION

See Section 5.6.52 of Ref. 7.

SUBROUTINE NAME: **ORTHLS**

DESCRIPTION

This subroutine finds the least squares polynomial which best approximates a weighted set of data points using orthogonal polynomials.

CALLING SEQUENCE

CALL ORTHLS (XA,YA,WA,IJ,LA,JA,CA,ALPHAA,BETAA,KA,TA,TB,TC,INDA)

where

XA	is the array of IJ data points for the independent variable.
YA	is the array of IJ data points for the dependent variable.
WA	is the array of IJ weights for the data points.
IJ	is the number of elements in the XA,YA, and WA arrays. Also, N is the number of elements in the TA, TB and TC arrays.
LA	is the weight switch: = 0 if weights are all set equal to one; = 1 if weights are input in th W array.
JA	is the number of low-order coefficients to be set equal to zero.
CA	is the array of KA + 1 computed polynomial coefficients.
ALPHAA	is the computed α array.
BETAA	is the computed β array.
KA	is the maximum degree of the polynomial to be fitted. Also, KA + 1 is the number of elements in the C array and K is the number of elements in the ALPHA and BETA arrays.
TA	is an array of IJ elements used for temporary storage by ORTHLS. Its contents upon return from ORTHLS are of no significance to the user.
TB	is an array of IJ elements used for temporary storage by ORTHLS. The contents upon return from ORTHLS are of no significance to the user.

TC	is an array of IJ elements for temporary storage by ORTHLS. The contents upon return from ORTHLS are of no significance to the user.
INDA	is the error indicator: = -1, when JA >KA = +1, when JA $\leq$ KA.

UTILITY ROUTINE AND COMMON REFERENCES

None.

METHOD OF SOLUTION

See Section 5.6.53 of Ref. 7.

SUBROUTINE NAME: **OUT**

DESCRIPTION

OUT writes the calculated data for data points along with the corresponding title and headings.

CALLING SEQUENCE

CALL OUT (I1,I2,K,K1W1,K1W2,FFPARY)

where 11,12 refer to the point numbers of the points to be output (any number of points may be output at one time. K represents the current normal line (takes on the value 1 or 2). PFPARY is the array containing the particle properties.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/
COMMON/DATAR/
COMMON/GASCON/
COMMON/HEAD/
COMMON/PARTP4/
COMMON/PARTP2/
COMMON/GAPPA/
COMMON/WRITPT/
COMMON/TEMPER/
COMMON/FTSTAG/
COMMON/CRITER/
COMMON/TOTAL/
COMMON/BOMOUT/
COMMON/CHEMCN/
COMMON/CHEMXX/
COMMON/GASDAT/
COMMON/UMIX2/
COMMON/VMIX3/
COMMON/PRAD/
COMMON/PSTR/
COMMON/AMF/
POFEM
PAGE
PFP
THERMO
PPATPT
NORSCK
VEMAG
SPCTX
ESHOCK
VMOD1
VMOD2

METHOD OF SOLUTION

Not applicable.

SUBROUTINE NAME: **OUTBIN**

DESCRIPTION

This subroutine writes the calculated normal data on the binary output tape. This is done for any number of data points. The routine will also punch (or write on a file) an exit plane start line or a restart startline.

CALLING SEQUENCE

CALL OUTBIN (I1,I2,JK,K1W1,K1W2,PFPARY)

where I1,I2 identifies the range of points to be written on tape (IJ1 is first point, I2 is last). JK represents the current characteristic line (1 or 2). K1W1 and K1W2 are flags which specify whether or not to punch a start line. PFPARY is the array containing the particle properties.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/TAPRIT/	COMMON/PSTR/
COMMON/DATAR/	COMMON/GAPPA/
COMMON/FORCE/	COMMON/GASDAT/
COMMON/GAPPA/	COMMON/PRAD/
COMMON/PARTP4/	COMMON/CHEMCN/
COMMON/PARTP2/	COMMON/CHEMXX/
COMMON/CONTRL/	COMMON/FSTAG/
COMMON/GLOBAL/	COMMON/VMIX3/
COMMON/AUX/	VMOD1
COMMON/AMF/	THERMO
COMMON/PUNEXT/	PFP
COMMON/FLAG/	TEMTAB
COMMON/ISTRT/	SPCTX
COMMON/TRPRT/	EMOFY

METHOD OF SOLUTION

Not applicable.

SUBROUTINE NAME: **OUTPUT**

DESCRIPTION

This subroutine prints out the calculated data for the data points in the single-phase transonic module.

CALLING SEQUENCE

```
CALL OUTPUT (L,PEPARY)
```

where L is the new timestep indicator and PFPARY is the array containing all the flow characteristics of the transonic data points.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATA1/
COMMON/SPEL/
PAGVOF
PHI
THERMT
MASS

METHOD OF SOLUTION

Not applicable.

SUBROUTINE NAME: **OVEREX**

DESCRIPTION

OVEREX solves for the shock angle at the nozzle lip when the flow is overexpanded. Provisions are made to calculate the shock angle for an upper or lower lip point. Real gas effects are considered in calculating flow properties downstream of the shock.

CALLING SEQUENCE

CALL OVEREX (PB,I,K,ITYPE1,K1W1,K1W2,PFPARY)

where PB is the freestream pressure at the boundary; I,K defines the location of the lip point in the characteristic data (PHO) array and ITYPE1 indicates whether an upper (=2) or lower (=1) boundary is to be considered. PFPARY is the array containing particle properties.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/	POFEM
COMMON/PARTP4/	ITSUB
COMMON/PARTP2/	PFP
EMOFV	UOFV
ESHOCK	IDMPFP
THERMO	ERRORS

METHOD OF SOLUTION

For the first pass through the solution, an initial shock angle is assumed. This shock angle is perturbed in ITUSB and the result used to calculate flow properties including static pressure downstream of the shock. The calculated static pressure is compared with the boundary pressure to determine if the desired convergence has been obtained. If the solution has not converged ITSUB is called again and the above procedure is repeated.

SUBROUTINE NAME: **PAGE**

DESCRIPTION

This subroutine page ejects and writes the header comments and page number on each page of printout.

CALLING SEQUENCE

```
CALL PAGE (LCNT,K1W1,K1W2)
```

where LCNT is a counter which monitors the number of lines of printed output per page. LCNT is reinitialized in PAGE.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/HEAD/
COMMON/CONTROL/
UTILITY - None

METHOD OF SOLUTION

When the maximum number of lines per page (55) have been output, PAGE is called to page eject. It then prints the identifying information and the page number, increments the page number and reinitializes the line counter.

SUBROUTINE NAME: **PAGVOF**

DESCRIPTION

The subroutine page ejects and writes the header comments on each page of printout for the single-phase transonic module.

CALLING SEQUENCE

CALL PAGVOF

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/HEAD/

METHOD OF SOLUTION

Not applicable.

SUBROUTINE NAME: PARINT

DESCRIPTION

The subroutine determines the particle properties for the portion of the first normal downstream of a Prandtl-Meyer expansion which is influenced by the expansion fan.

CALLING SEQUENCE

CALL PARINT (INL,I11,K,IFG,PFPARY)

where INL is the number of particle limiting streamlines which penetrate the Prandtl-Meyer fan, (I11,K) is the first point in the Prandtl-Meyer fan on the new line, IFG is a flag which is returned to the calling routine which is a zero if the intersection of the particle streamline was performed satisfactorily or a 10 if the intersection was above or below the expansion. If IFG returns as a 10 the new normal (K-line) is recalculated with a smaller step size. PFPARY is the array containing particle properties of the old and new normals.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/LSAD/	COMMON/TOTAL/
COMMON/PARTP2/	INRSCT
COMMON/PARTP4/	CHECK
COMMON/GAPPA/	PFP
COMMON/DATAR/	IDMPFP

METHOD OF SOLUTION

The Prandtl-Meyer expansion is performed assuming there are no particles present. This routine calculates the properties of any particles which enter the zone of influence of the Prandtl-Meyer fan. The limiting streamlines and associated properties which enter the fan are known before entering this routine (from PARSTR). Once the new normal has been completed, including the expansion fan points, PARINT uses the previously saved data to intersect the limiting streamlines with the new normal to locate the new limiting streamline and associated properties. The particle properties at all the remaining particle sizes at the other expansion fan points are determined via interpretation.

SUBROUTINE NAME: PARLOK

DESCRIPTION

The routine determines the intersection of the particle limiting streamlines with the edge of the boundary layer. It also determines the particle properties at the edge of the boundary layer so that PBLTRC and TRACEP can trace particle streamlines through the boundary layer.

CALLING SEQUENCE

```
CALL PARLOK(R9,ICT,NPAR,PFPARY)
```

where R9 is the radical coordinate of the boundary layer edge at each boundary layer solution station. R9 is the coordinate at which the boundary layer/boundary layer edge velocity is .995. ICT is the boundary layer station number, NPAR is the number of particle sizes which are found at any given station and PFPARY is the array which contains particle properties which are read from the inviscid nozzle solution tape and are used to determine the particle properties at the boundary layer edge.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DAT/	COMMON/SAVTEM/
COMMON/EVERY/	COMMON/SPEL/
COMMON/FLOW/	COMMON/STATN/
COMMON/ITOTI/	COMMON/TRCDAT/
COMMON/LIMIT/	READF
COMMON/PARTP2/	INTERP
COMMON/QUITIT/	TAPMOV

METHOD OF SOLUTION

The inviscid nozzle solution (saved on tape) and boundary layer solution (BLIMPJ) results are known prior to entry to PBLTRC. The boundary layer edge at each of the boundary layer solution stations is also known. PARLOK utilizes the inviscid nozzle flowfield tape to determine the particle properties at the edge of each boundary layer station along with the location and properties of all particle limiting streamlines which intersect the boundary layer anywhere inside the nozzle.

SUBROUTINE NAME: **PARSTR**

DESCRIPTION

This subroutine saves the particle streamline location and properties at the last normal before a Prandtl-Meyer expansion.

CALLING SEQUENCE

CALL PARSTR (INL,ITRY,J,K,ITST,PFPARY)

where INM is the number of particle limiting streamlines which enter the expansion fan, (ITRY,K) is the point number on the new normal which is being overriden by the Prandtl-Meyer expansion and whose properties and properties of the point above it are to be saved. J is the line identifier for the old normal prior to the expansion fan. ITST is a 1 if the particle limiting streamline is at the nozzle wall (i.e., particle impingement with the nozzle wall has occurred) and ITST is a 2 for all particle limiting streamlines which have not impinged on the nozzle wall. PFPARY is the array which contains the particle properties on the old and new normals.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAK/
COMMON/GAPPA/
COMMON/LSAD/
COMMON/PARTP2/
COMMON/PARTP4/
COMMON/TOTAL/
PFP

METHOD OF SOLUTION

This routine saves only the data for any particles size groups whose streamlines have been determined by PHASE1 to enter the zone of influence of the Prandtl-Meyer fan on the first normal downstream of the expansion corner. This information which is saved by PARSTR will subsequently be used by PARINT to determine particle flow properties in the Prandtl-Meyer zone of the first normal beyond the expansion corner.

SUBROUTINE NAME: **PARTIL**

DESCRIPTION

This subroutine is the control subroutines for the Approximate Transonic Analysis module. This routine and the other routines were taken from the SPP code (Ref. 7) and modified for use by the RAMP2F program to obtain the two-phase transonic start line.

CALLING SEQUENCE

CALL PARTIL (NTAPE,NSETS)

where

NTAPE	the FORTRAN unit on which the startline is stored (usually 8)
NSETS	the number of start line points for which there are particles present

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/ILINE/	COMMON/NAMEY/	COMMON/XXSH/	ORTHLS
COMMON/DELT/	COMMON/ERR/	COMMON/CONTRL/	COEFS
COMMON/DRAG/	COMMON/NAMEA/	COMMON/PCTC/	PAGE
COMMON/DATAR/	COMMON/NAME1/	PROP	TAB
COMMON/NAMEL/	COMMON/RZW1/	FIXIL	SITER
COMMON/NAMEM/	COMMON/SOURCE/	ONED	TOFEM
COMMON/NAMEQ/	COMMON/PARTP2/	DLTA	THERMO
COMMON/NAMER/	COMMON/FILIT/	WDGI	EMOFV
COMMON/NAMES/	COMMON/GASDAT/	TRACE	TOFV
COMMON/NAMEX/	COMMON/GASCON/		

METHOD OF SOLUTION

See Section 5.6.55 of Ref. 7.

SUBROUTINE NAME: **PARTIN**

DESCRIPTION

This subroutine reads in gas and particle property startline data. Data are read in from cards or tape.

CALLING SEQUENCE

```
CALL PARTIN (NSETS,NTAPE,PFPARY)
```

where

NSETS is the number of startline points where particles are present

NTAPE is the FORTRAN unit to read the startline data from (=7 for cards), NTAPE usually = 8 for two-phase and single-phase startline generated by the program, PFPARY is the array containing particle properties.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/PARTP4/	COMMON/VISEX/
COMMON/PART2/	COMMON/FSTAG/
COMMON/INPUT/	COMMON/BOMOUT/
COMMON/CONTRL/	COMMON/TRPRT/
COMMON/MASSC/	COMMON/SPEL/
COMMON/WT/	COMMON/CHEMXX/
COMMON/PSLD/	RGVOFM
COMMON/ONTSPT/	UOFEM
COMMON/GASCON/	TOFEM
COMMON/TEMPER/	POFEM
COMMON/NSF/	SPCTX
COMMON/LIPCOM/	THERMO
COMMON/PCTC/	IDMPFP
COMMON/GASDAT/	PFP

METHOD OF SOLUTION

The gas startline points are read starting with the axis point and input up to the boundary, while the particle startline data is input starting with the last limiting streamline or last gas startline point and input down to the axis.

SUBROUTINE NAME: **PARTPH**

DESCRIPTION

This subroutine reads and sets up the data table of particle temperature versus enthalpy. This routine also prints out the particle drag tables as well as the temperature versus enthalpy tables.

CALLING SEQUENCE

```
CALL PARTPH (IPFTOC,LCT,NGS)
```

where

IPFTOC = zero for two-phase case
= 10,000 for gas only case

LCT = line counter for printout purposes

NGS = a dummy variable

UTILITY ROUTINES AND COMMON REFERENCES

```
COMMON/GASTPE/
COMMON/CONTRL/
COMMON/TPHEH/
COMMON/GASDAT/
COMMON/DRAGCF/
COMMON/PARTP2/
COMMON/DATAR/
COMMON/TFLAG/
COMMON/TAPRIT/
COMMON/TRPRT/
PAGE
```

METHOD OF SOLUTION

Not applicable.

SUBROUTINE NAME: **PBLTRC**

DESCRIPTION

The subroutine provides overall control over the particle trajectory tracing module for determination of the properties of the particles in the boundary layer at the exit plane of the nozzle.

CALLING SEQUENCE

CALL PBLTRC (PFPARY)

where PFPARY is an array which is generally used to store particle properties but is used as a working array in the particle trajectory tracing module.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CHOCK/	COMMON/TFLAG/
COMMON/CONTROL/	COMMON/TRCDAT/
COMMON/CONVV/	IDMPDT
COMMON/GAPPA/	IDMPRO
COMMON/HITWAL/	INRSCP
COMMON/IPMX/	PARKOK
COMMON/MET/	PRO
COMMON/PARTP2/	PRO
COMMON/PRAD/	TAPMOV
COMON/PSLD/	TRACEP
COMMON/QUITIT/	WTFLOP
COMMON/SPEL/	WRITP

METHOD OF SOLUTION

This routine uses as input the inviscid nozzle results saving on unit 3 and the boundary layer results which are stored on unit 2. Particle properties in the boundary layer at the exit plume are determined via the following:

STEP 1 The boundary layer results are read in from unit 2, converted to vertical data stations along the nozzle wall and saved for further use.

STEP 2 The particle properties at the edge of the boundary layer are determined from the flowfield tape (unit 3) and saved for initializing the trajectory tracing. If no particles are found to penetrate the boundary layer then the routine is exited.

STEP 3 For each particle size that penetrates the boundary layer, eight particle trajectories (at even axial increments from the limiting streamline penetration to the exit plane) plus the limiting streamline trajectories are traced through the boundary layer up to the exit plane.

STEP 4 All particle sizes and properties are merged at the exit plane, summarized and stored for future use by the program for generating a viscous, two-phase startline for a plume station.

FUNCTION NAME: PDP

DESCRIPTION

This function computes the particle property data storage location and retrieving data from the PDAT array. This array contains the particle properties in the boundary layer at the exit plane of the nozzle.

CALLING SEQUENCE

= PDT(I,J,K,L)

where I,J,K,L are indices used to determine the storage location.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GAPPA/

METHOD OF SOLUTION

This routine simulates a four-dimensional array. The particle property storage location is computed using the following relation:

```
IX = I + 10 + (J-1 + 7*(K-1 + 10*(L-1)))
```

and retrieved using the relation

```
PDT = PDAT(IX)
```

FUNCTION NAME: **PFP**

DESCRIPTION

This function computes the particle property data storage location and retrieves data from the PFPARY array.

CALLING SEQUENCE

= PFP(I,J,K,L)

where

I,J,K,L are indices used to determine the storage location.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/PARTP3/
COMMON/PARTP2/
COMMON/PARTP4/

METHOD OF SOLUTION

The particle property data storage location is computed using the following relation

IX = I + 5 * (J-1 + 10 * (K-1 + 100 * (L-1));

and retrieved using the relation

PFP = PFPARY(IX).

SUBROUTINE NAME: **PHASE1**

DESCRIPTION

This subroutine provides the necessary controlling logic for the complete flowfield calculation. Proper subroutines are called to handle different kinds of calculations.

CALLING SEQUENCE

Call PHASE1 (IFINIS,ITRS,K1W1,K1W2,PFPARY)

where

IFINIS is a zero
K2W1,K2W2 are error flags
ITRS = 0 normal execution
= 10 two-phase solution with two-phase startline set up by program
= 20 exit plane startline using inviscid and viscous merged results
PFPARY is the array containing particle properties on the startline and subsequently used to store particle properties on the old and new normals.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/	COMMON/VMIX3/	POFEM
COMMON/DROP/	COMMON/VMIX4/	EMOFV
COMMON/GASCON/	COMMON/VMIX6/	RGMOFP
COMMON/NSF/	COMMON/PSTR/	VCFEM
COMMON/DATAR/	COMMON/CHEMCN/	THETPM
COMMON/GAPPA	COMMON/PUT/	TOFEM
COMMON/GLOBAL/	COMMON/AMF/	SOKSOL
COMMON/STEPC/	COMMON/PTEN/	STRNOR
COMMON/PARTP4/	COMMON/MASOUT/	ERRORS
COMMON/TEMPO1/	COMMON/CAPUR/	MASSCK
COMMON/TEMPO2/	COMMON/ISEA/	CHECK
COMMON/TOTAL/	COMMON/WRITIT/	PFP
COMMON/OVERLA/	COMMON/CUTFO/	IDMPFP
COMMON/CRITER/	COMMON/CPSV/	INRSCT
COMMON/INTEU/	COMMON/LIPFX/	SOKINT
COMMON/PSEC/	COMMON/IRN/	FREEMC
COMMON/TEMPO3/	COMMON/SKIPPY/	THERMO
COMMON/FREE/	COMMON/PUNEXT/	PRFRBD
COMMON/XXSH/	COMMON/LSAD/	EXPCOR
COMMON/BPRESW/	OUT	MESH
COMMON/PCTC/	THRUST	NORMCK
COMMON/CHEMXX/	OUTBIN	VMODEL

COMMON/VISEX/	SPCTX	VMOD1
COMMON/EXPER/	PPATPT	PLOAD
COMMON/GASDAT/	LIMITS	IBOUND
COMMON/FSTAG/	BOUND	INTT
COMMON/WAFT/	BOUNDA	AVERAG
COMMON/PARTP2/	PRANDT	IRAD
COMMON/LIPPT/	UOFV	SLSKIP
COMMON/BLMDAT/	ITERM	ALGINT
COMMON/RSTART/	TURN	CBREAK
COMMON/PRAD/	HYPER	PARSTR
		PARINT
		SLINT

METHOD OF SOLUTION

This subroutine makes most of the tests to determine what kind of calculation should be carried out for the point under consideration. The point may be a regular field point, solid or free boundary point, expansion corner points at the free boundary, and expansion corner points at solid boundary, etc. This routine calls either the characteristics or shock capturing methodology for solving points depending on user selected optio-..

FUNCTION NAME: **PHI**

DESCRIPTION

This function computes the flow property data storage location for any grid point at the new or old timestep in the single phase transonic module. This function returns the required data from the PFPARY which is used as a working array in this module.

CALLING SEQUENCE

= PHI(I,J,K,L,PFPARY)

where I,J,K,L are indices used to determine the storage location in the PFPARY array.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/

METHOD OF SOLUTION

This routine simulates a four-dimensional array. The flow property storage location is computed using the following relation:

IX = I + 12 * (J-1 + 15 * (K-1 + 27 * (L-1)))

and retrieved using the relation

PHI = PFPARY(IX).

SUBROUTINE NAME: PHYSOL

DESCRIPTION

This subroutine finds the reference properties on the characteristic line so that the compatibility equations can be used to calculate the flow velocity and angle of a point downstream of the known reference normal line (or surface).

CALLING SEQUENCE

```
CALL PHYSOL (PRET, IS, JS, IN, KN, IDIR, IFLAG, K1W1, K1W2, P1S,
PIN, PM, PM1, IPM, IPM1, KPM, JAG, P, ARGN, ISLIP, KSLIP, IF1X,
1141, IQUAD, H, SAVE, DP, PFPARY)
```

where

PRET(8)	is the storage array of reference properties found
(IS,JS)	is the point on the reference normal line (J-line), normally on the same streamline as the one under consideration
(IN,KN)	is a known point just below the point under consideration on the new normal line (K-line)
IDIR	indicates if a I-characteristic (=+1) or a II-characteristic (=-1) is being considered
IFLAG	is a control indicator to return the proper message to the calling subroutine in order that a proper measure can be taken
PIS(8)	array containing the flow properties of the streamline base point
PIN(8)	array containing the new flow properties of the streamline point
PM(8)	array containing the flow properties of point IPM which brackets the characteristic intersection
PMI(8)	array containing the flow properties of point IPM1 which brackets the characteristic intersection
IPM,IPM1	the point numbers of the two adjacent points on the old data surface which brackets the characteristic intersection

KPM	if the characteristic line intersects a boundary, shock, or slipline KPM is the point number on the new data surfa e which bounds the intersection
JAG	the point immediately above or below the streamline base point. This point is used to detect the presence of a slipline
P(8)	array in which the characteristic intersection flow properties are stored
ARGN	the angle of the normal
ISLIP	flag which indicates if not enough data is known to obtain the characteristic intersection
KSLIP	if KSLIP is a 1 the characteristic has intersected a slipline
IFIX	index used within PHYSOL which indicates if the two points which bracket the characteristic intersection have been found
I141	flag which indicates if the characteristic intersection is below the first point or above the last point on the old data surface
IQUAD	1 - interpolation is being made on R 2 - interpolation is being made on X
H	interpolation factor between point IPM and IPM1 necessary to obtain the characteristic intersection
SAVE(8)	array which is used to retain data from previous intersections
DP(8)	array which contains the flow property differences between points IPM and IPM1
PFPARY	array wnich contains particle properties.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/POINTC/	COMMON/TEMP02/	COMMON/CRITER/
COMMON/GASCON/	COMMON/PARTP4/	BOUND
COMMON/TEMPER/	COMMON/CPSV/	THERMO
COMMON/FSTAG/	COMMON/VMIX3/	INRSCT
COMMON/CHEMXX/	COMMON/CAPUR/	ITSUB
COMMON/CONTRL/	COMMON/PARTP2/	PFP
COMMON/DATAR/	COMMON/GAPPA/	UOFV
COMMON/SLIPPT/	COMMON/DROP/	GAPPB1
		PPATPT
		VMODEL

METHOD OF SOLUTION

The characteristic line is drawn from the point under consideration to intersect the known upstream reference normal line. The reference properties of this intersection are interpolated from the two known points on the reference normal line. Subroutine ITSUB and the average quantities are used to obtain a better approximation of the reference properties.

If the reference properties are not readily available, IFLAG is set to 2, and the reference properties are then assumed to enable the calculation to be continued. Normally, the calculation of this point is repeated afterward to obtain the correct reference properties for the calculation of the new point under consideration.

SUBROUTINE NAME: **PLMOUT**

DESCRIPTION

PLMOUT prints the data read by PLUMIN.

CALLING SEQUENCE

```
CALL PLMOUT (KP,LCNT,K1W1,K1W2,PFPARY)
```

WHERE KP is a control parameter set in PLUMIN, and LCNT is the printed line counter. PFPARY is the array which contains particle properties.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/	COMMON/WT/
COMMON/CUTFO/	COMMON/STEPC/
COMMON/GASDAT/	COMMON/WAFT/
COMMON/DATAR/	COMMON/FREE/
COMMON/GASCON/	COMMON/MOL/
COMMON/HEAD/	COMMON/TAPRIT/
COMMON/INPUT/	COMMON/FSTAG/
COMMON/GAPPA/	COMMON/PRAD/
COMMON/PARTP4/	PAGE
COMMON/PARTP2/	TAB
COMMON/MASSC/	IDMTAB
COMMON/PARTTP/	EMOFV
COMMON/PSLD/	THERMO
COMMON/DRAGCF/	PFP

METHOD OF SOLUTION

Not applicable.

SUBROUTINE NAME: **PLOAD**

DESCRIPTION

This subroutine loads data from one point into another point.

CALLING SEQUENCE

CALL PLOAD (I,K,L,J,MM,PFPARY)

where point (I,K) is being loaded with all the data defining point (L,J). MM is indicator (1,2) used to temporarily store gas specie mole fraction data for storage into the (I,K) point gas species storage location. PFPARY is the array which contains particle property information.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CHEMCN/
COMMON/CONTRL/
COMMON/DATAR/
COMMON/FSTAG/
COMMON/GLOBAL/
COMMON/PARTP2/
COMMON/PARTP4/
IDMPFP
PFP
SPCTX

METHOD OF SOLUTION

Not applicable.

SUBROUTINE NAME: **PLUMIN**

DESCRIPTION

PLUMIN reads in the input data (input via cards) necessary to perform the streamline-normal sclution. This routine provides control for all input functions by selectively calling pertinent input routines and/or the transonic solution.

CALLING SEQUENCE

```
CALL PLUMIN (K1W1,K1W2,NTAPE,NSETS,RRT,XSHSV,ITRS,PFPARY)
```

where

K1W1 K2W2 are multi-use flags

NTAPE is FORTRAN unit or tape number on which starting line is written

NSETS is the number of starting line points which have particles present

RRT is the nozzle throat radius of curvature ratio

XSHSV distance from nozzle throat to center (x = 0) of coordinate system

ITRS = 0 read in all input data except startline if read from tape
1 read in starting line from tape or FORTRAN unit NTAPE

PFPARY is the array which contains the startline particle properties.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/	COMMON/SIGNAL/	COMMON/GASDAT/
COMMON/CUTFO/	COMMON/LIPCOM/	COMMON/ACOM/
COMMON/DATAR/	COMMON/DRAGCF/	COMMON/IDL/
COMMON/GASCON/	COMMON/PSLD/	COMMON/EXPNN/
COMMON/HEAD/	COMMON/CRITER/	COMMON/KAPUR/
COMMON/INPUT/	COMMON/WAFT/	PARTPH
COMMON/STEPC/	COMMON/XXSH/	PLMOUT
COMMON/TFLAG/	COMMON/FREE/	IBOUND
COMMON/SIGMB/	COMMON/MOL/	STARTV
COMMON/VISEX/	COMMON/TAPRIT/	LOGIC
COMMON/VARSL/	COMMON/CHEMCN/	STLINE
COMMON/PARTTP/	COMMON/PRAD/	GASRD

COMMON/PARTP4/	COMMON/VMIX3/	BOUND
COMMON/PARTP2/	COMMON/TRPRT/	LIPIN
COMMON/GAPPA/	COMMON/SKIPPY/	AOASTR
COMMON/WRITPT/	COMMON/RSTART/	MASCON
COMMON/MASSC/	COMMON/ISTRT/	SETHTG
COMMON/BPRESW/	COMMON/STSVB/	PARTIN
COMMON/PARTWT/		

METHOD OF SOLUTION

Not applicable.

FUNCTION NAME: **POFEM**

DESCRIPTION

This function computes the local static pressure as a function of Mach number, entropy, and total temperature (ideal gas, two-phase only).

CALLING SEQUENCE

P = POFEM (EM,S,K1W1,K1W2)

where P is the resultant static pressure found from the Mach number, EM, and entropy, S. NOTE: Appropriate values of the gas properties must be stored in common upon entry to this routine.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON/
COMMON/TEMPER/
UTILITY - None

METHOD OF SOLUTION

Thermally perfect gas relationships are used to find the pressure.

$$p = p_o e^{-S/R} \left(1 + \frac{\gamma-1}{2} M^2\right)^{-\gamma/\gamma-1} \left(\frac{T_o}{T_c}\right)^{-\gamma/\gamma-1}$$

SUBROUTINE NAME: **POFH,PAFH***

DESCRIPTION

This routine utilizes the tabulated data of enthalpy and specific heat as functions of temperature for each species of a finite rate chemistry case to calculate pressure, as a function of enthalpy for a real gas, in a Prandtl-Meyer expansion.

CALLING SEQUENCE

```
CALL POFH (VF,HT,DELTA)
CALL PAFH(VF,HT,DELTA)
```

where

VF is the final velocity
HT is the total enthalpy
DELTA is the flow deflection angle.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON/
COMMON/PCTC/
COMMON/EXPER/
COMMON/CPMUK/
ITSUB

METHOD OF SOLUTION

The routine solves for pressure by incrementally changing the flow angle until the final flow angle is obtained. At each increment the routine determines new gas properties from the tables on enthalpy and specific heat as functions of temperature, then uses these properties for the next increment. The result is an integration of the flow properties through the angular change, DELTA.

*POFH and PAFH are the same routines but are included as two separate routines in the program because of overlay requirements and the need to keep the core requirements as small as possible.

SUBROUTINE NAME: **POINT**

DESCRIPTION

This subroutine is used by the single-phase transonic module to calculate the flow properties for interior and nozzle wall points.

CALLING SEQUENCES

```
CALL POINT (I,J,K,L,ITYPE,PFPARY)
```

where

I	point number on data surface
J	data surface number
K	old time step identifier (1 or 2)
L	new time step identifier (2 or 1)
ITYPE	1. Interior point for all but last downstream station. 2. Upper wall (nozzle) point for all but last downstream station. 3. Interior point for last downstream station. 4. Upper wall point for last downstream station.
PFPARY	array containing flow properties for the old and new time steps for all points in the transonic computational domain.

UTILITY ROUTINES AND COMMON REFERENCES

```
COMMON/DATAR/
PHI
WXANDR
WG
WTT
IDMPHI
```

METHOD OF SOLUTION

See Ref. 14.

FUNCTION NAME: **POP**

DESCRIPTION

This function computes the species mole fraction storage location and retrieves data from PFPARY. PFPARY has been temporarily loaded with the species mole fraction data which has been read from the TRAN72 thermodynamic tape which contains the equilibrium or equilibrium/frozen data for the particular propellant system which was stored originally on FORTRAN unit 10.

CALLING SEQUENCE

```
= POP(I,J,K,L,PFPARY)
```

where I,J,K,L are indices used to determine the storage location in the PFPARY array.

UTILITY ROUTINES AND COMMON REFERENCES

None.

DESCRIPTION

This routine simulates a four-dimensional array. The species mole fraction storage location is computed using the following relation:

```
IX = I + 10 * (J-1 + 2 * (K-1 + 25 * (L-1)))
```

and is retrieved using the relation

```
POI = PFPARY(IX).
```

SUBROUTINE NAME: **PPATPT**

DESCRIPTION

This subroutine calculates and stores gas and particle dependent variables as a function of the independent flow properties.

CALLING SEQUENCE

CALL PPATPT (M,IC,KC,VG,THETA,SG,K2W1,K2W2,KP,ISKIP,PG,PFPARY)

where

M	is the number of particle sizes present at the point
IC	is the point number for which particle and gas flow properties are to be calculated
KC	is the line identification flag
VG	is the gas velocity at the point
THETA	is the gas flow angle at the point
SG	is the gas entropy at the point
K2W1	is a dummy variable
K2W2	is a dummy variable
KP	is the temporary array storage location for the particle and gas flow properties
ISKIP	= 0 calculate particle properties only = 20 calculate gas and particle properties = 40 calculate gas properties only
PG	array containing the point independent flow properties
PFPARY	array containing the particle properties for all points.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/TFLAG/	COMMON/TEMPER/
COMMON/PARTP4/	COMMON/FSTAG/
COMMON/PARTP2/	COMMON/DATAR/
COMMON/GAPPA/	COMMON/FREE/
COMMON/ONTSPT/	COMMON/CRITER/
COMMON/GASCON/	COMMON/PSLD/
COMMON/CONTRL/	COMMON/XXSH/

COMMON/CPMUK/	POFEM
COMMON/DRAGCF/	PFP
COMMON/PCTC/	TEMTAB
COMMON/ICAPTUR/	DRAGMR
COMMON/PBLL/	DRAGCP
COMMON/VISEX/	

METHOD OF SOLUTION

The routine is entered knowing the gas independent variables (V,S,OF or H_T) and particle independent variables (u,v,ρ,h). The gas dependent variables (T,P,γ,μ, C_p,Pr) and particle dependent variables (R_E, drag and heat transfer terms) are calculated and stored for use in other parts of the code.

SUBROUTINE NAME: **PRANDT**

DESCRIPTION

This subroutine computes the Prandtl-Meyer expansion angle for a given boundary angle and divides this angle into a series of expansion "rays" (unless the number of rays has been specified in the input). The flow properties at each angular increment are set and stored in the PHO array.

CALLING SEQUENCE

```
CALL PRANDT (I,J,THETAB,NPM,IFLAG,ITYPE,K1W1,K1W2,PFPARY)
```

where

I	represents the corner point
J	indicates a characteristic line
THETAB	is the boundary angle
NPM	number of Prandtl-Meyer increments (calculated in PRANDT)
IFLAG	is an error flag
ITYPE	indicates if upper (2) or lower (1) boundary is being considered
PFPARY	is array containing particle properties for new and old normal

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CRITER/
COMMON/DATAR/
COMMON/GASCON/
COMMON/STEPC/
COMMON/CONTRL/
COMMON/PCTC/
common/CPMUK/
COMMON/PARTP4/
COMMON/PARTP2/
COMMON/FSTAG/
COMMON/CHEMXX/
COMMON/EXPER/
COMMON/VMIX3/

THERMO
THETPM
UOFV
EMOFV
TOFV
POFEM
TOFH
PAFH
SPCTX
VMODEL

METHOD OF SOLUTION

The routine is entered with known flow properties at the point of discontinuity along with the known corner and boundary flow angles. From the known angles and the preset number of degrees per ray, the number of increments is calculated. The distribution of P-M rays is then adjusted by a weighting function. Subroutine THETPM is entered with known initial conditions and the number of degrees per ray and returns with a velocity. These new conditions are then set into the PHO array. See Volume I, Sections 6 and 7.7, for the details of calculation.

SUBROUTINE NAME: **PREAD**

DESCRIPTION

The subroutine reads the particle data from FORTRAN unit or tape unit 12 on which the inviscid exit plane start line has previously been stored. This data will be used to construct a new startline at the exit plane that includes the nozzle boundary layer.

CALLING SEQUENCE

```
CALL PREAD (PFPARY,NSETS)
```

where PFPARY is the array in which the particle data will be stored and NSETS is the number of points on the exit plane normal which contain particles.

UTILITY ROUTINES AND COMMON REFERENCES

```
COMMON/CONTRL/
COMMON/PARTP2/
COMMON/PARTP4/
IDMPFP
```

METHOD OF SOLUTION

Data are read in starting with the points nearest the lip and proceeding to the axis point.

SUBROUTINE NAME: **PRFRBD**

DESCRIPTION

This subroutine calculates the flow properties at the intersecion of a particle limiting streamline and a plume boundary.

CALLING SEQUENCE

```
CALL PRFRBD (IS,JS,IN,KN,I,K,PFPARY)
```

where

IS	=	point number of the old (JS) data surface plume boundary
JS	=	line indicator of the old data surface
IN	=	point indicator of the old (JS) data surface limiting streamline
KN	=	line indicator of the old data surface
I	=	point number of the new (K) data surface limiting streamline
K	=	line indicator of the new data surface.
PFPARY	=	the array containing the particle properties at old and new normal.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/	COMMON/CONTRL/
COMMON/PARTP4/	COMMON/FSTAG/
COMMON/PARTP2/	PFF
COMMON/GAPPA/	INRSCT
COMMON/SLIPPT/	IDMPFP
COMMON/ONTSPT/	PPATPT

METHOD OF SOLUTION

When the new data surface has been completed and it has been determined that a particle limiting streamline has crossed the plume boundary, the location of the intersection is determined by the intersection of a line passing through the old and new limiting streamline points. This establishes two interpolation factors. One along the limiting streamline and one along the plume boundary. Gas properties at the intersection point are interpolated for between the two plume boundary points and particle properties are interpolated for between the two limiting streamline points. The interpolated point and properties are then used as the plume boundary point for the new line, and the calculation for the next line is then initiated.

SUBROUTINE NAME: PRO

DESCRIPTION

This function computes the boundary layer flow property data storage location. This function returns the boundary layer flow properties from the PFPARY which is used as a working array in the particle trajectory tracing module.

CALLING SEQUENCE

= PRO(I,J,K,PFPARY)

where I,J,K are indices used to determine the storage location in the PFPARY array.

UTILITY ROUTINES AND COMMON REFERENCES

None.

METHOD OF SOLUTION

This routine simulates a three-dimensional array. The boundary layer flow property storage location is computed using the following relation:

$$IX = I + 14 * (J-1 + 15 * K-1))$$

and retrieved using the relation

$$PRO = PFPARY(IX).$$

SUBROUTINE NAME: PROP

DESCRIPTION

This subroutine uses subroutine HALL to calculate the gas properties at any point in the transonic region for a two-phase nozzle.

CALLING SEQUENCE

```
CALL PROP(I)
```

where

I = 0 for the initial call to ONED
1 for subsequent calls

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/	COMMON/NAMEX/	COMMON/RZMAP/
COMMON/NAMEA/	COMMON/NAMEY	COMMON/PW1/
COMMON/NAMEL/	COMMON/UBVB/	COMMON/NAMEW/
COMMON/NAMER/	COMMON/NAME1/	COMMON/RZW1/
COMMON/NAMES/	COMMON/RWTD/	HALL

METHOD OF SOLUTION

See Section 5.6.62 of Ref. 7.

SUBROUTINE NAME: **PUNEX**

DESCRIPTION

This subroutine generates the input data required by the JANNAF Standard Plume Flow Field (SPF) program to start its solution at the exit plane of a rocket nozzle.

CALLING SEQUENCE

CALL PUNEX (PFPARY)

where PFPARY is the array which normally contains particle property data for the old and new normals but is used as a temporary working routine by PUNEX.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CAPUR/	COMMON/TPEH/
COMMON/CHEMCN/	COMMON/TRPRT/
COMMON/CONTRL/	COMMON/VISEX/
COMMON/CPMUK/	COMMON/FISSS/
COMMON/DATAR/	COMMON/VMIX3/
COMMON/GAPPA/	COMMON/SKIPPY/
COMMON/GASDAT/	COMMON/SPECI/
COMMON/ISTRT/	COMMON/SPEL/
COMMON/PARTP2/	ARASSL
COMMON/PARTP4/	IDMPOP
COMMON/PARTWT/	SPECIE

METHOD OF SOLUTION

The exit plane flowfield information at the exit plane has been previously stored by Phase 1 and SLSKIP on FORTRAN unit 12 during the inviscid nozzle solution. PUNEX reads the flow properties from unit 12 orders the points from the axis to the lip and punches or reloads (on unit 12) the SPF startline information. If no boundary layer is to be calculated and only inviscid SPF startline data are required, then PUNEX will punch the SPF data in the proper format. If a boundary layer is to be generated by the BLIMPJ code then the SPF startline data is stored on FORTRAN unit 12 so that an inviscid/viscous startline can be generated by subroutine BLEXIT in a subsequent RAMP execution. At the present time PUNEX will output data for any of the chemical systems specified in Volume I, Section 2. The viscosity, Prandtl number, single specie distribution, particle specific heat, and particle density are output as exit plane mass flow averages. This can easily be changed to reflect increased capabilities of subsequent versions of the SPF code. PUNEX will output both single- and two-phase SPF startlines.

SUBROUTINE NAME: **RREADF**

DESCRIPTION

The subroutine reads the gas and particle properties of a single normal from the flowfield output tape (Unit 3). This routine is used by the particle trajectory tracing module to load data necessary to specify particle properties at the edge of the boundary layer.

CALLING SEQUENCE

```
CALL READF (ITOT,J,LINE,PFPARY)
```

where

ITOT	=	the number of points on the normal which have been read.
J	=	the line indicator for the line just read (1 or 2)
LINE	=	the number of normals that have been read from the flowfield tape
PFPARY	=	the array in which the particle properites for each normal (line) is stored.

UTILITY ROUTINES AND COMMON REFERENCES

```
COMMON/CHEMCN/
COMMON/CONTRL/
COMMON/CONVV/
COMMON/DAT/
COMMON/EVERY/
COMMON/ITOTT/
COMMON/LIMIT/
COMMON/MET/
COMMON/PRAD/
IDMPFP
```

METHOD OF SOLUTION

Not applicable.

FUNCTION NAME: **RGMOFP**

DESCRIPTION

This subroutine finds Mach number as a function of pressure, O/F ratio (or total enthalpy) and entropy. The difference between this routine and EMOFP is that in this case the gas properties are not known prior to entry.

CALLING SEQUENCE

```
EM = RGMOFP (OF,S,P,K2W1,K1W1)
```

where EM is the resultant Mach number, P is the local static pressure, S is the local entropy, and OF is the local O/F ratio (or total enthalpy).

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/	ITSUB
COMMON/GASCON/	VOFEM
COMMON/ISEA/	EMOFP
COMMON/GASDAT/	ERRORS
POFEM	THERMO
EMOFV	

METHOD OF SOLUTION

The real gas tables have, as independent variables, OF ratio (total enthalpy), entropy and velocity. If the velocity is not known, an iterative solution must be employed to find Mach number from pressure, entropy, and OF ratio (or total enthalpy).

FUNCTION NAME: **RGVOFM**

DESCRIPTION

This subroutine finds velocity as a function of Mach number, entropy and O/F ratio (or total enthalpy). The difference between this routine and VOFEM is that the gas properties are not known prior to entry.

CALLING SEQUENCE

```
V = RGVOFM (OF,S,EM,K2W,K1W)
```

where V is the resultant velocity computed from O/F ratio or total enthalpy, OF, entropy, S, and Mach number, EM.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/	TAB
COMMON/CHEMCN/	FOVEM
COMMON/GASDAT/	EMOFV
COMMON/GASCON/	ITSUB
THERMO	ERRORS

METHOD OF SOLUTION

The real gas tables have, as independent variables, OF ratio (or total enthalpy), entropy, and velocity. If the velocity is not known, an iterative solution must be employed to find the velocity from Mach number, OF ratio (or total enthalpy) and entropy.

FUNCTION NAME: **RHOFEM**

DESCRIPTION

RHOFEM computes the local density as a function of Mach number and entropy.

CALLING SEQUENCE

RHO = RHOFEM (EM,S,K1W1,K1W2)

wehre RHO is the resultant density found from local Mach number and local entropy. NOTE: The appropriate values of the gas properties must be stored in common upon entry to this routine.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON/
POFEM

METHOD OF SOLUTION

Thermally perfect gas relationships are used to find the density.

$$\rho = \rho_o \left(1 + \frac{\gamma-1}{2} M^2\right)^{-(1/\gamma-1)}$$

SUBROUTINE NAME: **RITE**

DESCRIPTION

This subroutine tells the program user (in no uncertain terms) that he has made a fatal error. The next executable statement is a STOP.

CALLING SEQUENCE

```
CALL RITE(I)
```

UTILITY ROUTINES AND COMMON REFERENCES

None.

METHOD OF SOLUTION

Not applicable.

FUNCTION NAME: **ROTERM**

DESCRIPTION

ROTERM computes the geometrical factor, F_I,F_{II}, used in the axisymmetric term of the compatibility equation and as an interpolation parameter.

CALLING SEQUENCE

F = ROTERM (THETA,DELTA,EMU,R3,RI,K2W1,K2W2)

where

THETA is the flow angles of the known points (θ_I or θ_{II})
DELTA defines the quadrant being considered
EMU is the Mach angles of the known points (μ_I or μ_{II})
R3 is the coordinates of the new point ($\bar{r}_{III}$ or $\bar{x}_{III}$)
RI is the coordinate of the known point (r_I or x_I)

UTILITY ROUTINES AND COMMON REFERENCES

None.

METHOD OF SOLUTION

The method-of-characteristics solution uses this routine to determine a coefficient needed in its solution. This term (see Eq. (6.29), Section 6 of Ref. 4) can be written as:

$$F = \frac{\left|\sin\mu\right| (d_{III} - d)}{\sin(\pi/4 + \delta(\bar{\theta} + \bar{\mu} - \pi/4))}$$

By the proper choice of d(r or x), δ and the sign of μ, indeterminant forms are eliminated in the evaluation.

SUBROUTINE NAME: **SETHTG**

DESCRIPTION

This routine computes the gas total enthalpy for a case when finite rate chemistry is being used and the startline is to be generated by the program for gaseous flows only.

CALLING SEQUENCE

CALL SETHTG

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/VISEX/	COMMON/LIPCOM/
COMMON/PCTC/	COMMON/SIGMB/
COMMON/GASCON/	COMMON/GASDAT/
COMMON/CHEMCN/	TKEY
COMMON/CHEMXX/	THERMO

METHOD OF SOLUTION

The routine interpolates for the flow properties at the specified startline Mach number using the equilibrium thermodynamic data tables. The resultant temperature and velocity are then used to obtain the flow properties from the species enthalpy and specific heat tables. The total enthalpy is calculated from the static enthalpy and velocity. This procedure is used to ensure property compatibility when transferring from the equilibrium tables to the species finite rate tables.

C-3

SUBROUTINE NAME: **SITER**

DESCRIPTION

This routine determines the entrop, of the gas knowing the velocity, static pressure and total enthalpy of O/F ratio.

CALLING SEQUENCE

```
CALL SITER (HG,S,EM,V,PC,PL,IFLG,ICV)
```

where

HG is the known total enthalpy or O/F ratio
S is the gas entropy
EM is the gas Mach number
V is the known gas velocity
PC is the gas total pressure
PL is the known gas static pressure
IFLG = 0 Velocity is known on entering the routine
= 1 Mach number is known on entering routine
ICV = 0 routine has converged
= 1 no convergence

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON/
EMOFV
ITSUB
POFEM
THERMO
RGVOFM

METHOD OF SOLUTION

This subroutine iterates on the gas entropy until the guessed entropy, known velocity, and enthalpy results in a static pressure which are within the convergence criteria of the known static pressure.

SUBROUTINE NAME: SLDP

DESCRIPTION

This subroutine finds the solutions to a set of N simultaneous linear equations.

CALLING SEQUENCE

```
CALL SLDP(X,A,N)
```

where

X is the solution matrix
A is the coefficient matrix
N is the order of the coefficient matrix.

UTILITY ROUTINES AND COMMON REFERENCES

None.

METHOD OF SOLUTION

The set of N simultaneous equations are solved using a Gauss-Jordan reduction scheme with the diagonal pivot strategy.

SUBROUTINE NAME: **SLINT**

DESCRIPTION

This subroutine determines if a particle limiting streamline is sufficiently close to the upper boundary (nozzle wall or free boundary) to specify that the limiting streamline has impinged on the wall or crossed the free boundary.

CALLING SEQUENCE

```
CALL SLINT(K,J1,PFPARY)
```

where

K	is the new line indicator
J1	is a 0 if no impingement is determined is a 1 if the limiting streamline has impinged the wall or passed through the plume boundary
PFPARY	is the array which contains the particle properties at the old and new normal.

UTILITY ROUTINES AND COMMON REFERENCES

```
COMMON/CONTRL/
COMMON/DATAR/
COMMON/FSTAG/
COMMON/GAPPA/
COMMON/PARTP2/
COMMON/PARTP4/
COMMON/TOTAL/
PPATPT
SPCTX
UOFEM
```

METHOD OF SOLUTION

If a particle limiting streamline is adjacent to the upper boundary and within 0.5 percent of the radial position of the boundary point then the particles are assumed to impinge on the boundary. The old limiting streamline point below the boundary point is deleted and the particle properties at the limiting streamline are assumed to apply at the upper boundary. Subsequent normals downstream of this point are assumed to have particles of this size group present and any mass hitting the boundary is assumed to stick to the wall or pass through a free boundary. If a boundary layer solution is performed after the inviscid nozzle is calculated and an exit plane startline is to be generated with the boundary layer and particle trajectory tracing then the user can specify an accommodation coefficient which allows all or some of the impinged particle mass flux to be carried out into the plume solution. For more detail see Section 7.6 of Volume I.

SUBROUTINE NAME: SLSKIP

DESCRIPTION

This subroutine saves the exit plane data for use in setting up the SPF startline.

CALLING SEQUENCE

CALL SLSKIP(IS,J,I,K,PFPARY)

where

(IS,J) is the streamline point on the old normal
(I,K) is the corresponding streamline point on the new normal
PFPARY is the array containing particle properties on the old (J) and new (K) normals.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CHEMCN/
COMMON/CHEMXX/
COMMON/CONTRL/
COMMON/DATAR/
COMMON/FSTAG/
COMMON/GASDAT/
COMMON/PARTP2/
COMMON/PARTP4/
COMMON/PARTP4/
COMMON/SKIPPY/
COMMON/TFLAG/
PFP
SPCTX
TEMTAB

DESCRIPTION

Once PHASE1 has determined that a streamline has crossed the exit plane SLSKIP is called with the streamline points of the (IS,J) point upstream of the exit plane and the (I,K) point downstream of the exit. The flow characteristics at the exit point are interpolated for and written on FORTRAN unit 12. The data on unit 12 will be used by other routines (PUNEX) when the nozzle solution has been completed.

SUBROUTINE NAME: SPACET

DESCRIPTION

This subroutine is used to space past the nozzle flowfield data on the flowfield data tape (Unit 3).

CALLING SEQUENCE

CALL SPACET(LIN)

where LIN is the number of normals read from the flowfield tape to get to the exit plane of the nozzle.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/MET/

DESCRIPTION

The routine is called only when an exit plane restart is being made using the data generated by a previous nozzle run and the boundary layer data generated by the BLIMPJ code. This routine is used to space past the nozzle data on the flowfield data file so that the plume data may be stored on the same file.

SUBROUTINE NAME: SPCTX

DESCRIPTION

This routine is used to retrieve and store the species mole fraction data.

CALLING SEQUENCE

```
CALL SPCTX(IFCN,IPT,ILINE,JLINE)
```

where

IFCN indicates to store (=1) data into the species array or to retrieve (=2) data from the species array and store it in a temporary array.

IPT is the flowfield point number

ILINE is the line number (1 or 2)

JLINE specifies which temporary array location (SPCT(I,1) or SPCT(I,2)) to use to store or retrieve data.

UTILITY ROUTINES AND COMMON REFERENCES

```
COMMON/CHEMXX/
COMMON/CHEMCN
COMMON/CONTRL/
COMMON/SPEL/
```

DESCRIPTION

The RAMP code uses a temporary array (SPCT(25,2)) for storage of the species mole fraction information for finite rate chemistry cases. This subroutine retrieves and stores information from/into a permanent array. When the routine is used to store information in the permanent array (SPE(I,IPT,ILINE)) it uses data from the SPCT(I,JLINE) array. When the code retrieves the information from the permanent array (SPE(I,IPT,ILINE)) it stores it in the temporary array (SPCT(I,JLINE)).

FUNCTION NAME: **SPECC**

DESCRIPTION

This function computes the species mole fraction storage locations from the array which contains the equilibrium chemistry thermodynamic results. This routine is used to determine the species mole fractions used in setting up the BLIMPJ input data by the boundary layer data generation module (BLMPIN).

CALLING SEQUENCE

```
= SPECC (I,J,K,L)
```

where I,J,K,L are indices used to determine the storage location in the temporary SPECX.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/SPEL/

METHOD OF SOLUTION

This function simulates a four-dimensional array. Species mole fraction data have previously been stored in a temporary array (SPECX) as a function of total enthalpy, entropy, velocity, and species name. The appropriate storage location for the mole fractions is computed using the following relation

```
IX = I + 2 * (J-1 + 2 * (K-1 + 13 * (L-1)))
```

and retrieved using the relation

```
SPECC = SPECX(IX).
```

SUBROUTINE NAME: SPECIB

DESCRIPTION

This function determines the species mole fractions from the equilibrium thermodynamic data for use by the boundary layer input data module in setting up the BLIMPJ data for a equilibrium case.

CALLING SEQUENCE

CALL SPECIB (HOF,K,V,S)

where

HOF is the interpolation factor for the total enthalpy

K is the equilibrium chemistry total enthalpy table number to be used in the interpolation

V is the velocity at which the species mole fractions are desired

S is the entropy at which the specie mole fractions are desired.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/
COMMON/GASDAT/
COMMON/PARTP2/
COMMON/SPEL/
COMMON/SPFK/
TOB
SPECC

DESCRIPTION

This routine is used by the boundary layer input data module to determine the species mole fractions as a function of total enthalpy, entropy and velocity. This routine is only required for equilibrium chemistry cases where the thermodynamic data come from the TRAN72 program. The routine locates where in the thermodynamic table to find the corresponding species mole fractions and retrieves them from an array in which they have been previously stored.

SUBROUTINE NAME: **SPECIE**

DESCRIPTION

This routine determines the species mole fractions and transport properties from the equilibrium thermodynamic data for use by the SPF input data module.

CALLING SEQUENCE

```
CALL SPECIE (INU,ISK,OF,S,V,PFPARY)
```

where

INU	is the point number on the exit plane SPF startline
ISK	is the number of species in the particular chemical system
OF,S,V	is the total enthalpy (or O/F ratio), entropy and velocity of the gas at the particular point in the exit plane
PFPARY	is the particle array in which the species mole fractions are temporarily stored.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/	COMMON/PARTP2/
COMMON/CPMUK/	COMMON/VISSS/
COMMON/DATAR/	POP
COMMON/GAPPA/	TAB
COMMON/GASDAT/	THERMO

DESCRIPTION

This routine performs a function similar to SPECIB except that it is used by the SPF input data generation module (PUNEX). This routine determines the species mole fractions, and transport properties at each SPF exit plane startline point as a function of total enthalpy, entropy and velocity. This routine is only used for equilibrium chemistry cases where the thermodynamic data comes from the TRAN72 program. The routine locates where in the thermodynamic table to find the corresponding species mole fractions and transport properties, then it retrieves the data from the PFPARY in which the species mole fractions have been previously stored in the correct order for the particular SPF chemical system. This routine is entered with the enthalpy, entropy, and velocity whereas SPECIB is entered with the interpolation factor for the total enthalpy, the total enthalpy table number, entropy, and velocity. SPECIE and SPECIB could be combined into a single routine, but for computer storage limitations and overlay requirements this has not been done.

SUBROUTINE NAME: START

DESCRIPTION

This routine determines the particle flow characteristics required to initiate the particle trajectory tracing through the boundary layer at the point where the particles enter the boundary layer.

CALLING SEQUENCE

```
CALL START (X,IST,I,P1)
```

where

X is the axial coordinate of the boundary layer edge where the particular particle trajectory is to start

IST is the particle size group number

I is the particular trajectory number (1-10)

P1 is the array in which the particle flow properties are returned ($V, \theta, H, T, \rho, Y, X$).

UTILITY ROUTINES AND COMMON REFERENCES

```
COMMON/SPEL/
COMMON/TRCDAT/
```

DESCRIPTION

A set of tables defining the particle properties at the boundary layer edge as a function of boundary layer station position has been previously set up by the particle trajectory tracing module. The particle trajectories are not necessarily initiated at a boundary layer station position. This routine interpolates for the particle properties at the edge of the boundary layer using the axial coordinate of the penetration of the boundary layer to look up data.

SUBROUTINE NAME: **STARTV**

DESCRIPTION

This subroutine sets up the grid and initializes the single phase transonic solution.

CALLING SEQUENCE

CALL STARTV(PFPARY)

where PFPARY is the array which normally contains particle properties but is temporarily used by the single-phase transonic module as a working array.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/	BOUND	OFSET
COMMON/DATAR/	IBOUND	OUTPUT
COMMON/GASCON/	IDMPHI	PHI
COMMON/SPEL/	ITSUB	POFEM
COMMON/STSUV/	MASS	RGVOFM
AOASTR	MAXT	RHOFEM

METHOD OF SOLUTION

The routine is entered knowing the combustion chamber and nozzle contour, an overall O/F ratio, the O/F distribution at the initial transonic starting plane (normally the entrance to the convergent section of the combustion chamber), the axial position to start the transonic solution, the axial coordinate of the throat and an area ratio at which to terminate the transonic system. Also known are the thermodynamic properties of the system. The following steps are taken to initialize the single-phase transonic solution:

Step 1: The mass flow of the motor is determined either as an input or using the overall O/F ratio, the throat area (using the axial coordinate of the throat) and the requirement that the flow be sonic at the throat.

Step 2: The axial coordinate of the downstream cutoff for the transonic module is determined using the input downstream cutoff area ratio or the default area ratio of 1.5.

Step 3: The computational grid is set up in the Cartesian coordinate system. Ten solution stations are equally spaced up to and including the throat and the starting station. Downstream of the throat 17 equally spaced stations are set up out to the downstream station of step 2. At each axial station equally spaced points are set up including the wall and centerline.

Step 4: The O/F ratio at each of the points on the upstream boundary surface is set using the input O/F versus radial position distribution. The remaining flowfield points O/F distribution is initially set the same as the initial data surface. The flow angle at each point is assumed to vary linearly from the axial to the nozzle wall at each station.

Step 5: The remaining flow variables (P,T,V,H) are determined by iterating using pressure (which is the independent variable used to look up temperature, velocity, molecular weight, and gamma from the thermodynamic tables) to force the mass flow through the particular data plane to match the mass flow calculated in step 1. The code is forced to take the appropriate solution (subsonic or supersonic depending on which side of the throat the initialization is taking place) for the pressure.

Step 6: The maximum allowable time step for each point in the computational domain is then calculated using the following relation:

$$DT = \frac{DTCUT * DR}{(M+1)\sqrt{8 * a^2}}$$

where

DTCUT is the factor to cut step size (.2)

DR is the radial distance between two mesh points

M is the Mach number

a is the sonic velocity.

Step 7: The maximum step size for the overall solution is then determined by using MAXT to search for the smallest time step of al computational points.

Note: The maximum number of radial points is 15(ICON(2)).

SUBROUTINE NAME: **STGMOD**

DESCRIPTION

This subroutine computes the gas thermodynamic properties in the transition flow regime.

CALLING SEQUENCE

```
CALL STGMOD(I,K)
```

where

I = the point number
K = the line number

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON/
COMMON/FREE/
COMMON/GSV/
COMMON/FSTAG/
COMMON/DATAR/
COMMON/TEMPER/
UTILITY - None

METHOD OF SOLUTION

The routine is entered knowing the flow regime, Knudsen number and flow properties (M_w,T,P,V,U,S,H,γ) of the (I,K) point. The specific heat ratio is then determined based on the flow regime.

Continuum - is the same as entered
Vibrational mode frozen - γ is set to 1.4
Rotational mode frozen - γ is set based on a curve fit of gamma from 1.4 (vibrationally frozen) to 1.667 (free molecular) based on Knudsen number
Translationally frozen (free molecular) - γ = 1.667.

Once the local gamma is determined then the local static properties, T, P, and V, are used to determine the local total conditions (T_o,P_o) and Mach number. While provisions are made for handling transitional flow at the present time this routine is called only for translationally frozen flow.

SUBROUTINE NAME: STLINE

DESCRIPTION

This routine stores the input gaseous properties of startline from a temporary array used by the input module into the permanent array used by the flowfield computational module.

CALLING SEQUENCE

CALL STLINE

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/
COMMON/DATAR/
COMMON/INPUT/

METHOD OF SOLUTION

Not applicable.

SUBROUTINE NAME: **STRNOR**

DESCRIPTION

This subroutine handles the calculation of the flow properties of the point in question. The following cases are considered:

1. Interior point, uses I- and II-characteristic (ITYPE = 11)
2. Lower solid boundary point, uses I-characteristic (ITYPE = 21)
3. Upper solid boundary point, uses II-characteristic (ITYPE = 22)
4. Lower free boundary point, uses I-characteristic (ITYPE = 31)
5. Upper free boundary point, uses II-characteristic (ITYPE = 32)

Except those ITYPE numbers shown above, sometimes, one of the following numbers (500, 600, 700, 800, 900) is added to the original number to transmit more information to this subroutine.

CALLING SEQUENCE

```
CALL STRNOR (I1,K1,IS1,JS1,IN1,KN1,IFLAG,ITYPE,K1W1,K1W2)
```

where

I1,K1 is the storage location in the PHO array for the point in question on the new normal (K-line)

IS1,JS1 is the storage location in the PHO array for the known reference point on the old normal (J-line); normally this point is on the same streamline as the point I1,K1

IN1,KN1 is the storage location in the PHO array for the known point I1,K1 on the new normal (K-line)

IFLAG is a control indicator for sending in and out necessary messages

ITYPE denotes the type of point to be calculated.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/FREE/	COMMON/STEPC/	PPATPR
COMMON/ISEA/	COMMON/AVPROP/	FNEWTN
COMMON/CHEMCN/	COMMON/PARTP4/	INRSCT
COMMON/CHEMXY/	COMMON/PARTP2/	GAPPBI
COMMON/CPSV/	COMMON/GAPPA/	PHYSOL
COMMON/LIPNT/	COMMON/ONTSPT/	SPCTX
COMMON/VMIX3/	COMMON/POINTC/	PFP
COMMON/AVPRP2/	COMMON/NSF/	RGMOFP
COMMON/CAPUR/	COMMON/OVERLA/	COEFF3

COMMON/LIPFX/	COMMON/CPMUK/	ROTERM
COMMON/CONTRL/	COMMON RUE/	SLPLIN
COMMON/CRITER/	COMMON/CHEMXX/	NEWENT
COMMON/DATAR/	COMMON/TUIPA/	UOFV
COMMON/DROP/	COMMON/GLOBAL/	VOFEM
COMMON/GASCON/	COMMON/PSEC/	COEFEQ
COMMON/PHISOL/	COMMON/INTEU/	CHECK
COMMON/SLIPPT/	COMMON/PARSTU/	VMODEL
COMMON/TEMP02/	COMMON/FSTAG/	EMOFV
COMMON/TOTAL/	BOUND	SITER
COMMON/CP[illegible]SS/	IDMPFP	

METHOD OF SOLUTION

Initially the flow properties of the point in question are assumed to be the same as those of the known upstream point on the same streamline, and its location is found by intersecting the average streamline from the reference point (IS1,JS1) on the J-line and the average normal from the known point (IN1,KN1) on the K-line. Subroutine PHYSOL is used to find the reference properties for the characteristic lines and Eq.(4.135a) and (4.100) of Vol. I are then used to calculate velocity and flow angle for the new point.

The iterative method is employed to find the velocity and the flow angle until they do not change appreciably between the successive iterations. During this iteration, the location of the new point is perturbed.

FUNCTION NAME: TAB

DESCRIPTION

This function computes the thermodynamic data storage location and retrieves data from the TABB array.

CALLING SEQUENCE

= TAB (I,J,K,L)

where

I,J,K,L are the indices which are used to determine the storage location

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASDAT/
UTILITY - None

METHOD OF SOLUTION

This routine simulates a four-dimensional array. The thermodynamic data storage location is computed using the following relation

IX = I + 10 * (J-1 + 2 * (K-1 + 13 * (L-1)))

and retrieved using the relation

TAB = TABB(IX)

SUBROUTINE NAME: TAPMOV

DESCRIPTION

This routine spaces past the header, thermodynamic data, and particle thermodynamic data on the flowfield data file (unit 3). This routine is used by the particle trajectory tracing module to get to the beginning of the flowfield data on the output tape.

CALLING SEQUENCE

CALL TAPMOV

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/
COMMON/MET/

METHOD OF SOLUTION

The header and transport data are read from unit 3 until the flowfield data are reached on the file.

SUBROUTINE NAME: **TEMTAB**

DESCRIPTION

This subroutine will perform a table lookup for particle temperature as a function of enthalpy or for particle enthalpy as a function of temperature.

CALLING SEQUENCE

```
CALL TEMTAB(X,Y,WHICH)
```

where

X is the unknown variable

Y is the known variable

WHICH is the lookup control variable indicating to lookup temperature (=1) or enthalpy (=2).

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTROL/
COMMON/TPEH/
COMMON/TFLAG/
UTILITY - None.

METHOD OF SOLUTION

The unknown variable (particle temperature or enthalpy) is calculated by either assuming constant heat capacities or by applying linear interpolation techniques to the tabulated data input on cards 33.

SUBROUTINE NAME: **THERMO**

DESCRIPTION

This subroutine utilizes real or ideal gas information obtained from the flowfield tape (or tables) and a local O/F ratio (or total enthalpy) to call subroutine FABLE to calculate thermodynamic gas properties locally in the flow.

CALLING SEQUENCE

```
CALL THERMO (OF,SS,VV)
```

where

OF = gas total enthalpy or O/F ratio
SS = gas entropy
VV = gas velocity.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON/	COMMON/FAB/
COMMON/CPMUK/	COMMON/CONTRL/
COMMON/GRINT/	COMMON/VLTM/
COMMON/PARTP4/	TAB
COMMON/PARTP2/	FABLE
COMMON/GASDAT/	THERM1
	THERMV

METHOD OF SOLUTION

The routine is entered with the local O/F ratio (or total enthalpy), OF, entropy, SS, and velocity, VV. The local ratio is used to determine which set of thermodynamic tables that subroutine FABLE should use to perform table lookup of the local thermodynamic gas properties. Subroutine THERMO then uses the local thermodynamic gas properties obtained from FABLE to perform an interpolation between the O/F (or total enthalpy) tables based on the local O/F ratio (or total enthalpy). If a finite rate case is being calculated then THERM1 is called to compute the local thermodynamic gas properties. This routine also checks to see if the velocity which is entered is beyond the limiting velocity of either of the two tables bounding in local O/F ratio (or total enthalpy). If the limiting velocity is exceeded the THERMV is called to calculate the local thermodynamic gas properties.

SUBROUTINE NAME: THERMT

DESCRIPTION

This routine determines the gas thermodynamic properties for the single phase transonic module.

CALLING SEQUENCE

CALL THERMT

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/
COMMON/GASDAT/

METHOD OF SOLUTION

This routine performs the same function as THERMO and is used to determine the local thermodynamic properties of the gas. Unlike THERMO which uses O/F (or total enthalpy), S and V to look up thermo data, THERMT uses O/F (or total enthalpy) and pressure to look up the thermo data. The thermodynamic tables which are used by THERMT are the same as those used by THERMO except they are organized so that pressure is the dependent variable.

SUBROUTINE NAME: **THERMV**

DESCRIPTION

This routine determines the gas thermodynamic properties for equilibrium chemistry cases for which the local velocity has been found to exceed the limiting velocity of either of the two tables used to determine the gas thermodynamics.

CALLING SEQUENCE

```
CALL THERMV(OF,S,VV,S1,S2)
```

where

OF = gas total enthalpy or O/F ratio
S = gas entropy
V = gas velocity
S1 = reference entropy level of first (high total pressure) entropy table at the local OF
S2 = reference entropy level of second (low total pressure) entropy table at the local OF.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTROL	COMMON/TEMPER
COMMON/CPMUK/	COMMON/VLIM/
COMMON/GASCON/	COMMON/VTRY/
COMMON/GASDAT/	EMOFV
COMMON/MOL/	POFEM
COMMON/PARTP2/	TOFV

METHOD OF SOLUTION

The RAMP code like the MOC (Ref. 12) code uses precalculated tables of gas thermodynamic data which are generated by the TRAN72 program. The use of these tables is explained in Appendix A of Volume I and Ref. 13 (MOC). The tables are generated for specific O/F ratios (or total enthalpy) and entropy levels (total pressure). Each table is made up of a series of velocity entries corresponding to points on an isentropic expansion from the total conditions for the particular table. At each velocity in the table the thermodynamic properties are tabulated. Since there are limitations to how many tables of O/F ratio (10) and entropy (2) which can be handled by the program it is possible that the local gas velocity might exceed the limiting velocity of one of the tables being used for the interpolation.

Subroutine THERMV is used when a gas limiting velocity is exceeded. Subroutine THERMO calculates local properties for each individual table at the entered velocity, then calculates the equivalent total conditions. THERMT calculates total conditions for the last entry of each table then interpolates for total conditions at the desired O/F (or enthalpy) for each entropy table. The local thermodynamic properties at each entropy table are then calculated. Another interpolation is performed based on the entropy level to get the local thermodynamic properties for the entered O/F, S, and V. Finally, total conditions corresponding to the particular state of the gas (T, P, O/F, S, V) are calculated for use by other portions of the program.

SUBROUTINE NAME: **THERM1**

DESCRIPTION

This routine determines the gas thermodynamic properties for a finite rate chemistry case.

CALLING SEQUENCE

```
CALL THERM1(HT,V)
```

where

HT is the gas total enthalpy
V is the gas velocity.

UTILITY ROUTINES AND COMMON REFERENCES

```
COMMON/TEMPER/
COMMON/GASCON/
COMMON/PCTC/
COMMON/CPMUK/
COMMON/CONTRL/
COMMON/VISEX/
COMMON/CHEMCN/
COMMON/CHEMXX/
COMMON/GASDAT/
TKEY
TOFH
```

METHOD OF SOLUTION

The routine looks up enthalpy and specific heats from tabulated data of enthalpy and specific heats as functions of temperature. The enthalpy, specific heats and molecular weights of each species are used, along with species concentrations, to calculate the mixture gas constant, gamma, enthalpy, specific heat, total pressure, and temperature. These properties, along with velocity are used to calculate total enthalpy and Mach number.

SUBROUTINE NAME: **THETPM**

DESCRIPTION

THETPM performs a numerical integration to calculate properties through a Prandtl-Meyer expansion. Either the case of known final velocity or known final expansion angle may be handled.

CALLING SEQUENCE

CALL THETPM (OF,S,DELTA,VF,CI,IT,ITYPE,K1W,K2W)

where

OF	is the local O/F ratio or total enthalpy
S	is the local entropy level
DELTA	is the total expansion angle
VF	is the final velocity downstream of the expansion
VI	is the initial velocity upstream of the expansion
IT	is a control parameter indicating if expansion to a solid wall or free boundary is taking place
ITYPE	indicates if an upper (2) or lower (1) boundary is being considered.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON	TOFH
COMMON/STEPC/	ITSUB
COMMON/CONTRL/	TOFV
THERMO	ERRORS

METHOD OF SOLUTION

The integral equation

$$\int_{V_1}^{V_F} \sqrt{M^2 - 1}\,\frac{dV}{V} - \Delta\theta = f(V_F) = 0$$

where $M^2 = V^2/\gamma RT$ is solved knowing either the final velocity, V_F, or the expansion angle ($\Delta\theta$). As can be seen, if the final velocity, V_F, is known, the integration progresses straightforwardly to a solution. However, if the expansion angle is known, an iterative procedure must be employed to pick the velocity which produces the desired expansion angle.

SUBROUTINE NAME: **THRUST**

DESCRIPTION

THRUST computes the vacuum thrust produced by a two-dimensional or axisymmetric nozzle. Addition of the thrust at the throat and the integrated pressure along the nozzle wall yields the final thrust.

CALLING SEQUENCE

CALL THRUST (L,K,I1,J1,ITYPE,ICALC,K1W1,K1W2,PFPARY)

where L,K designates the unknown characteristic point and I1, J1 is the known characteristic point. ITYPE specifies if the point is on the upper or lower boundary and ICALC is a counter with the values of 1, 2, or 3. (1 specifies integration at the throat, 2 - along the nozzle and 3 - at the exit.) PFPARY is the array which contains particle properties on the old and new normals.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/
COMMON/DATAR/
COMMON/FORCE/
COMMON/PARTP4/
COMMON/PARTP2/
COMMON/FSTAG/
COMMON/WT/
COMMON/PSLD/
COMMON/INTCR/
COMMON/TRPRT/
PFP
VEMAG

METHOD OF SOLUTION

Thrust is found by first computing the (gas and particle) momentum thrust in the sonic area or throat of the nozzle. The static pressure is then integrated along the nozzle wall and the total thrust found by summing of the pressure and momentum terms (both gas and particle). Inclusion of a factor in the incremental force term accounts for either two-dimensional or axisymmetric flow.

SUBROUTINE NAME: TKEY

DESCRIPTION

This routine determines the proper index to be used in the enthalpy and specific heat tables and calculates interpolation factors.

CALLING SEQUENCE

```
CALL TKEY (T,TTB,ITKEY,SDT,HDT,NT)
```

where

T = the temperature
TTB = the temperature tables used as independent variables
ITKEY = the resultant index
SDT and HDT = interpolation factors
NT = number of entries in the temperature table.

UTILITY ROUTINES AND COMMON REFERENCES

None.

METHOD OF SOLUTION

The routine searches the temperature table until the input temperature is bounded. The index of the lower bound is stored in ITKEY and the interpolation factors are calculated by the equations

$$SDT = \frac{T - TTB\ (ITKEY)}{TTB\ (ITKEY + 1) - TTB\ (ITKEY)}$$

and

$$HDT = \frac{TTB\ (ITKEY + 1) - T}{TTB\ (ITKEY + 1) - TTB\ (ITKEY)}.$$

FUNCTION NAME: TOFEM

DESCRIPTION

TOFEM computes the local static temperature as a function of Mach number. TOFEM and TOFV are quite similar; the difference being if Mach number or velocity is the known quantity.

CALLING SEQUENCE

T = TOFEM (EM,K1W1,K1W2)

where T is the one-dimensionally calculated local static temperature which exists at the Mach number, EM. NOTE: The appropriate values of the gas properties must be stored in common upon entry to this routine.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON/
UTILITY - None.

METHOD OF SOLUTION

The thermally perfect gas relationships are used to find the static temperature at the local Mach number.

$$T = \frac{T_o}{1 + \frac{\gamma-1}{2} M^2}.$$

FUNCTION NAME: TOFENH

DESCRIPTION

This routine calculates the temperature as a function of enthalpy for a finite rate chemistry case.

CALLING SEQUENCE

= TOFENH(HU)

where HU is the static enthalpy.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON/
COMMON/PCTC/
COMMON/CPMUK/
COMMON/VISEX/
COMMON/CHEMCN/
COMMON/CHEMX/
COMMON/GASDAT/
TKEY
ITSUB

METHOD OF SOLUTION

The temperature is estimated initially and this temperature is used to calculate an enthalpy from the temperature-enthalpy tables. If the resultant enthalpy does not match HU, the temperature is incremented and the process repeated until the enthapies converge.

FUNCTION NAME: TOFH, TAFH*

DESCRIPTION

This routine calculates the temperature as a function of enthalpy for a finite rate chemistry case during a Prandtl-Meyer expansion.

CALLING SEQUENCE

```
= TOFH (HU,V)
= TAFH (HU,V)
```

where

HU is the enthalpy
V is the velocity.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/TEMPER/
COMMON/GASCON/
COMMON/PCTC/
COMMON/CPMUK/
COMMON/VISEX/
COMMON/CHEMCN/
COMMON/CHEMXX/
COMMON/GASDAT/
TKEY
ITSUB
POFH

METHOD OF SOLUTION

The methodology is the same as for TOFENH except that the gas constant, molecular weight, gamma, and Mach number are also computed.

*TOFH and TAFH are the same routine but are included as two separate routines in the program because of overlay requirements and the need to keep the core requirements as small as possible.

FUNCTION NAME: TOFV

DESCRIPTION

This function computes the local static temperature as a function of velocity. TOFV and TOFEM are quite similar; the difference being if Mach number or velocity is the known variable.

CALLING SEQUENCE

T = TOFV (V,K1W1,K1W2)

where T is the local static temperature which exists at the velocity, V.
NOTE: The appropriate values of the gas properties must be stored in common upon entry to this routine.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON/
RITE
ERRORS
KIKOFF

METHOD OF SOLUTION

The thermally perfect gas relationships are used to find the static temperature at the local velocity.

$$T = T_o - \frac{V^2}{2R}\left(\frac{\gamma-1}{\gamma}\right).$$

SUBROUTINE NAME: **TRACE**

DESCRIPTION

This subroutine is used by the two-phase transonic module to integrate particle trajectories through the nozzle throat.

CALLING SEQUENCE

Call TRACE.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DRAG/	COMMON/NAMES/
COMMON/DATAR/	COMMON/NAMEX/
COMMON/NAMEL/	COMMON/NAMEY/
COMMON/NAMEM/	COMMON/NAMEA/
COMMON/NAMEQ/	COMMON/RZW1/
COMMON/NAMER/	PROP

DESCRIPTION

See Section 5.6.81 of Ref. 7.

SUBROUTINE NAME: **TRACEP**

DESCRIPTION

This routine solves the equation of motion of a particle along with its thermal response. This routine is used by the particle trajectory tracing module to trace particle trajectories through the nozzle boundary layer and predict the particle flow properties which exist at the nozzle exit plane in the boundary layer.

CALLING SEQUENCE

```
CALL TRACEP (P1,P2,RP,RHOP,DT,EP,ACC,XEND,PFRARY)
```

where

P1	array which contains the particle flow characteristics and location at the edge of the boundary layer
P2	array which contains the particle flow characteristics and location in the boundary layer at the exit plane of the nozzle
RP	particle radius
RHOP	particle mass density
DT	integration time step
EP	particle emissivity
ACC	particle accomodation factor
XEBD	axial location of exit plane of nozzle
PFPARY	particle property array which is temporarily used to store the spatial variation of boundary layer properties along the nozzle wall.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CHOCK/	COMMON/SLOW/
COMMON/CONVV/	DRAGCP
COMMON/DRAGCF/	DRAGMR
COMMON/FIND/	GAS
COMMON/SAVE/	TEMTAB

METHOD OF SOLUTION

This routine is called with the location and properties of a given particle size at the edge of the nozzle boundary layer. The equations of motion and energy equations are solved as the particle trajectory is traced through the known boundary layer flow field up to the exit plane of the nozzle. The routine also keeps track of which particles, which trajectory and where any nozzle wall impingement takes place. The properties of the particle at the exit plane or point of impingement on the wall are returned to the calling routine.

SUBROUTINE NAME: **TRANS**

DESCRIPTION

This subroutine provides overall control for initializing the data and reading the namelist data for the Kliegel two-phase transonic solution of a supersonic gas particle startline.

CALLING SEQUENCE

```
CALL TRANS (NTAPE,NSETS,RUT)
```

where

NTAPE = FORTRAN unit on which the startline is written
NSETS = number of startline points where particles are present
RUT = throat radius (ft).

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONTRL/	COMMON/PCTC/
COMMON/GASDAT/	COMMON/VISEC/
COMMON/GASCON/	COMMON/VSON/
COMMON/CPMUK/	COMMON/DATAR/
COMMON/TPEH/	COMMON/NAMER/
COMMON/MASSC/	COMMON/NAMEX/
COMMON/DRAGCF/	COMMON/NAMEQ/
COMMON/PARTP2/	COMMON/NAMEL/
COMMON/ERR/	COMMON/DRAG/
COMMON/NAMEA/	COMMON/R2W1/
COMMON/NAME1/	COMMON/RWTD/
COMMON/NAMEW/	TAB
COMMON/HUL/	PARTIL
	THERMO

METHOD OF SOLUTION

Not applicable.

SUBROUTINE NAME: **TURN**

DESCRIPTION

TURN solves for a shock wave which has a known turning angle (δ). A condition of known turning angle exists when the flow is turned through a compression corner on a solid boundary. Real gas effects are considered in calculating conditions downstream of the shock.

CALLING SEQUENCE

```
CALL TURN (PU,PD,DELTA,IFLAG,K1W1,K1W2)
```

where PU,PD represent flow conditions upstream and downstream of the shock, DELTA is the turning angle, and IFLAG indicates if the solution will or will not converge.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/BOMOUT/
COMMON/CRITER/
COMMON/CONTRL/
THERMO
EMOFV
UOFEM
ESHOCK
ITSUB
ERRORS
UOFV

METHOD OF SOLUTION

An initial shock angle is assumed. This shock angle is used to calculate a turning angle. The calculated turning angle is compared to the known turning angle and successive iterations on shock angle are performed until the turning angle difference is sufficiently close to zero.

FUNCTION NAME: **UOFEM**

DESCRIPTION

This function computes the local Mach angle as a function of local Mach number. Prior to the calculation, a test is made to ensure that the Mach number is greater than one.

CALLING SEQUENCE

```
EMU = UOFEM (EM,K1W1,K1W2)
```

where EMU is the Mach angle which exists at the local Mach number, EM.

UTILITY ROUTINES AND COMMON REFERENCES

```
COMMON/BOMOUT/
FPRORS
K_.OFF
R?1E
```

METHOD OF SOLUTION

The following equation is solved for the local Mach angle.

$$\mu = \tan^{-1}\left(\frac{1}{\sqrt{M^2 - 1}}\right).$$

FUNCTION NAME: **UOFV**

DESCRIPTION

This function computes the local Mach angle as a function of local velocity.

CALLING SEQUENCE

```
EMU = UOFV (V,K1W1,K1W2)
```

where EMU is the Mach angle which exists at the local velocity, V.

UTILITY ROUTINES AND COMMON REFERENCES

```
COMMON/BOMOUT/
UOFEM
EMOFV
```

METHOD OF SOLUTION

The local velocity is converted into a Mach number using EMOFV. Function UOFEM is then entered with the calculated Mach number. The Mach angle is obtained from the following equation.

$$\mu = \tan^{-1}\left(\frac{1}{\sqrt{M^2 - 1}}\right)$$

FUNCTION NAME: **VEMAG**

DESCRIPTION

VEMAG determines the magnitude of a vector.

CALLING SEQUENCE

= VEMAG(V)

where V is any vector.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON - None
DOTPRD

METHOD OF SOLUTION

The following equation is solved for the magnitude of a vector

$$\text{VEMAG} = \sqrt{V(1)^2 + V(2)^2}$$

where

$$\vec{V} = V(1)\,\vec{i} + V(2)\,\vec{j}.$$

FUNCTION NAME: **VOFEM**

DESCRIPTION

This function computes velocity as a function of Mach number.

CALLING SEQUENCE

V = VOFEM (EM,K1W1,K1W2)

where V is the local velocity which corresponds to the local Mach number, EM. NOTE: The appropriate values of the gas properties must be stored in common upon entry to this routine.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON/
TOFEM

METHOD OF SOLUTION

The thermally perfect gas relationship

$$V = \sqrt{\frac{R\,\gamma(T_o - T)}{\frac{\gamma-1}{2}}}$$

is solved for velocity. Local static temperature, T, is obtained from the input Mach number.

SUBROUTINE NAME: **WALPRP**

DESCRIPTION

This subroutine determines the distribution of points along the nozzle wall which will be used as boundary layer solution stations for the BLIMPJ code.

CALLING SEQUENCE

CALL WALPRP

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/WALPR/

METHOD OF SOLUTION

This routine uses the nozzle wall points and associated pressure distribution to establish the solution stations and pressure distribution which will be used by the boundary layer input module.

The nozzle wall is described using 34 stations. The first station in the nozzle wall point on the startline and the last point is the nozzle lip. The next three points are specified at distances away from the preceding point on the nozzle of 1/200th, 1/100th, and 1/50th of the distance from the first to the last points on the nozzle wall. The remaining points are equally spaced along the nozzle out to the lip. The appropriate flow properties that the boundary layer input module requires are interpolated from the inviscid nozzle wall results. This point distribution was chosen because it produced the best results for all cases that have been run using the boundary layer option.

SUBROUTINE NAME: **WDGI**

DESCRIPTION

This subroutine integrates along the initial two-phase supersonic startline for the mass flow by trapezoidal rule.

CALLING SEQUENCE

CALL WDGI

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/
COMMON/NAMEL/
COMMON/NAMEM/
COMMON/NAMES/
COMMON/RZW1/
PROP

METHOD OF SOLUTION

See Section 5.6.87 of Ref. 7.

SUBROUTINE NAME: WEAK, WEKK*

DESCRIPTION

This subroutine determines the independent variables, entropy and velocity, SD, VD, downstream of a weak oblique shock. The gas properties upstream of the shock are known prior to entry.

CALLING SEQUENCE

```
CALL WEAK (OF,SU,VU,EPS,DELTA,SD,VD,K1W,K2W)
```

where OF is the upstream O/F ratio (or total enthalpy), SU,VU are the upstream entropy and velocity, EPS, DELTA are the shock angle and turning angle, and SD,VD are the downstream entropy and velocity.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON/
THERMO
EMOFV
POFEM
RHOFEM
ENTROP
DELTAF

METHOD OF SOLUTION

From the known upstream entropy and velocity, the local gas properties, pressure, density, and upstream Mach number are calculated. The entropy rise across the shock is added to the upstream entropy to get total downstream entropy. Downstream velocity is calculated from the following relationship.

$$V_D = \frac{V_U \cos(\xi)}{\cos(\xi - \delta)} .$$

*WEAK and WEKK are the same routine but are included as two separate routines in the program because of overlay requirements and the need to keep the core requirements as small as possible.

SUBROUTINE NAME: WG

DESCRIPTION

This subroutine decodes the pressure, velocity, O/F ratio and density at each point in the single phase transonic module.

CALLING SEQUENCE

Call WG

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/

METHOD OF SOLUTION

See Ref. 14.

FUNCTION NAME: WOFA

DESCRIPTION

WOFA computes the weight flow per unit area as a function of Mach number. This calculation is used only in function AJASTR.

CALLING SEQUENCE

Weight Flow = WOFA (EM,K1W1,K1W2)

where EM is the local Mach number. NOTE: The appropriate values of the gas properties must be stored in common upon entry to this routine.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/GASCON/
UTILITY - None

METHOD OF SOLUTION

Weight flow per unit area, W/A, is calculated from the thermally perfect gas relation.

$$\frac{W}{A} = \sqrt{\frac{\gamma}{RT_o}} \left\{ \frac{P_o M}{\left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{\gamma+1}{2(\gamma-1)}}} \right\}.$$

SUBROUTINE NAME: WRITP

DESCRIPTION

This subroutine prints out a header and summarizes the particle flow properties as they enter the nozzle boundary layer as well as at the exit plane of the nozzle in the nozzle boundary layer.

CALLING SEQUENCE

```
CALL WRITP(IWR)
```

where

IWR = 1 print out particle properties at entrance to boundary layer
= 2 print out particle properties in boundary layer at exit plane of the nozzle

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/CONVV
COMMON/GAPPA/
COMMON/HITWAL/
COMMON/IPMX/
IDMPDT
PDT

METHOD OF SOLUTION

Not applicable.

FUNCTION NAME: **WTFLOF**

DESCRIPTION

This function computes the area normal to the flow which is bounded by two streamline points.

CALLING SEQUENCE

= WTFLOF (M,N,K,A)

where

M = the point number of the lower streamline point
N = the point number of the upper streamline point
K = the line number
A = a 1 for axisymmetric flow and a 0 for two-dimensional flow.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR/
COMMON/CONTRL/
UTILITY - None

METHOD OF SOLUTION

The area bounded by two points and normal to the average local flow vector is calcualted via geometric relations.

SUBROUTINE NAME: **WTFLOP**

DESCRIPTION

This routine is used by the particle trajectory tracing module to calculate the particle density distribution in the boundary layer at the exit plane of the nozzle.

CALLING SEQUENCE

```
CALL WTFLOP (IPMAX)
```

where IPMAX is the number of particle size groups which penetrate the boundary layer.

UTILITY ROUTINES AND COMMON BLOCKS

COMMON/ACOM/
COMMON/CONTRL/
COMMON/GAPPA/
COMMON/HITWAL/
COMMON/PARTP2/
COMMON/TRCDAT/
PDT
IDMPDT

METHOD OF SOLUTION

A mass flow balance is performed for each two consecutive particle trajectories to determine the density at each trajectory point at the exit plane of the motor. Any particle mass which impinges on the nozzle wall is applied to the nozzle lip point using a user input accommodation coefficient and an average particle velocity and temperature.

SUBROUTINE NAME: WTT

DESCRIPTION

This routine is used by the single phase transonic module to solve for pressure, velocity, and O/F ratio at each new point.

CALLING SEQUENCE

CALL WTT

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/DATAR

DESCRIPTION

See Ref. 14.

SUBROUTINE NAME: **WXANDR**

DESCRIPTION

This routine solves reference equations for single phase transonic routines.

CALLING SEQUENCE

```
CALL WXANDER (IN,JN,KN,ITYPE,PFRARY)
```

where

IN	point number on data surface
JN	data surface number
KN	old time step identifier (1 or 2)
ITYPE	1. Interior point for all but last downstream station. 2. Upper wall (nozzle) point for all but last downstream station. 3. Interior point for last downstream station. 4. Upper wall point for last downstream station.
PFRARY	Array contains flow properties for the old and new time steps for all points in the transonic computational domain.

UTILITIES AND COMMON REFERENCES

```
COMMON/DATAR/
PHI
```

METHOD OF SOLUTION

See Ref. 14.

FUNCTION NAME: XSI

DESCRIPTION

This function computes the storage location for the nonlinear interpolation weighting functions required for thermodynamic property lookup and retrieves data from XSIDIM.

CALLING SEQUENCE

= XSI (I,J,K,L)

where

I,J,K,L are indices used to determine the storage location.

UTILITY ROUTINES AND COMMON REFERENCES

COMMON/XSICOM/
UTILITY - None

METHOD OF SOLUTION

This routine simulates a four-dimensional array. The storage location is computed using the following relation

$$IX = I + 10 * (J-1 + 2 * (K-1 + 13 * (L-1)))$$

and retrieved using the relation

XSI = XSIDIM(IX).

5.2 DESCRIPTION OF TRAN72 SUBROUTINES

This subsection presents a brief description of all the subroutines that make up the TRAN72 program.

- BLOCK DATA

BLOCK DATA contains atomic data stored in ATOM(i,j) and many of the variables used with the variable format, FMT. The format variables are stored in the common labeled OUPT and are described here.

A variable format was used so that one format, FMT, could be used in the final output with changes in the number of decimal places according to the sizes of the numbers. The format is used to print a label and from 1 to 13 associated numbers. The labels contain 14 alphanumeric characters stored in four words and printed with 3A4,A2. The numbers are all printed in a field of 9. FMT is initially set in BLOCK DATA as follows:

```
FMT  (1)   (2)   (3)  (4)  (5) (6) (7) (8)  (9) (10) (11) (12) (13) (14) (15)
     (1H,  34A,  A2,  F9.  0,  F0. 0,  F9.  0,  F9.  0,   F9.   0,  F9.  0 ,

FMT (16) (17) (18) (29) (20) (21) (22) (23) (24) (25) (26) (27) (28) (29) (30)
     F9.   0,  F9.  0,   F9.  0,   F9.   0,  F9.   0,  F9.  0,   F9.   0,    )
```

where the spaces are stored as blanks.

Some variables set in BLOCK DATA to modify FMT are as follows:

```
Variable:  F0   F1   F2   F3   F4   F5   FB   FMT13   FMT9X   FMT19
Storage:   0    1,   2,   3,   4,   5,        13       9X,      I9,
```

The following is a list of variables used as labels and printed with 3A4, A2 in FMT:

Variable	Stored Label
FP	P,ATM
FT	T.DEG K
FH	H.CAL G
FS	S. CAL/(G)(K)
FM	M.MOL WT
FV	(DLV DLP)T
FD	(DLV DLT)P
FC	CP.CAL/(G)(K)
FG	GAMMA (S)
FL	SON VEL.M SEC
FR1	PC/P
FC1	CF
FN	MACH NUMBER
FR	CSTAR, FT/SEC
FI	ISP, LB-SEC/LB
FA	IVAC, LB-SEC/LB
FA1,FA2	AE/AT

- Subroutine CPHS

Subroutine CPHS calculates thermodynamic properties for species numbering from 1 to NS for an assigned temperature TT. It uses either one of two sets of coefficients: COEF(2,i,j) for the temperature interval TLOW to TMID and COEF(1,i,j) for the interval TMID to THIGH. The index j is the j^{th} species and the "index" i (i = 1 to 7) refers to the i^{th} coefficient.

The properties calculated and their corresponding FORTRAN symbols are as follows:

Property	FORTRAN Symbol
$(S_T^o/R)_j$	S(j) j = JS1,...,NS
$(H_T^o/RT)_j$	HO(j) j = JS1,...,NS
$\sum_{j=JS1}^{NS} n\ (C\ /R)$	CPSUM
$n_j(C_p^o/R)_j$	CPSUM (when CPHS Is called by HCALC).

The index JS1 is always set equal to 1 in all routines calling CPHS except HCALC. In the latter event JS1 and NS are both set equal to the index j of a particular species.

- Subroutine EFMT

Subroutine EFMT (E-format) writes statements in a special exponent form. This form is similar to the standard FORTRAN E-format, but the letter E and some of the spaces have been removed for compactness. It is used to write density and mole fractions with the TRACE option.

- Subroutine EQLBRM

EQLBRM is the control routine for the equilibrium module which calculates equilibrium compositions and thermodynamic properties for a particular point. The subroutine flow diagram is given in Figs. 5-1a through 5-1c. Figure 5-1a is a diagram of the complete routine. Figure 5-1b is an expansion of the block labeled 85 in Fig. 5-1a. It gives the details for obtaining and applying corrections in subroutine EQLBRM. Figure 5-1C is the expansion of the block between blocks labeled 160 and 143 in Fig. 5-1a. Figure 5-1c is the flow diagram for adding and removing condensed species.

Several common variables must be set before EQLBRM is called. These variables and the modules setting them are summarized below:

Module	Variables
Input	ENN, ENNL, IQ1, IUSE(1), JSOL, JLIQ, EN(j,NPT), ENLN(j) } for first iteration, first point only COEF(k,1,j), TEMP(j,k), IONS, LIMT(l), RR, NS NLM, NC, SHOCK, IDEBUG, TRACE, and TMID.
	For TP, HP, SP, TV, UV, and SV problems, TP, HP, SP, VOL, and S0 (if SP is true) are all set.
Application	TT, PP, NPT, VLM(NPT) (if VOL is true). In addition, if not previously set in Input: VOL, TP, HP, SP, and S0 (if SP is true).
Additional input processing	B0, EQRAT, and HSUB0

ORIGINAL PAGE IS
OF POOR QUALITY

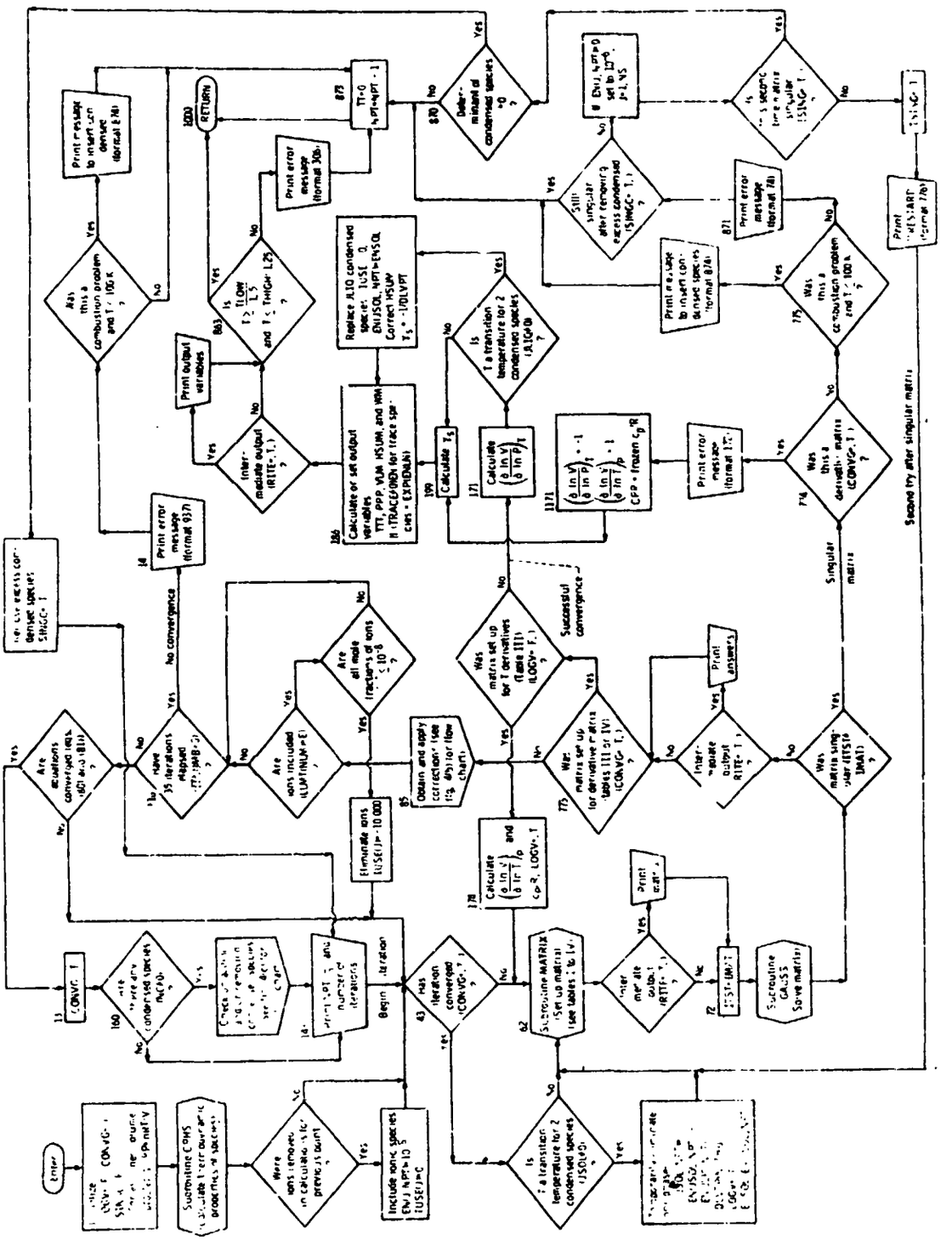


Fig. 5-1a - Flow Diagram for Subroutine EQLBRM

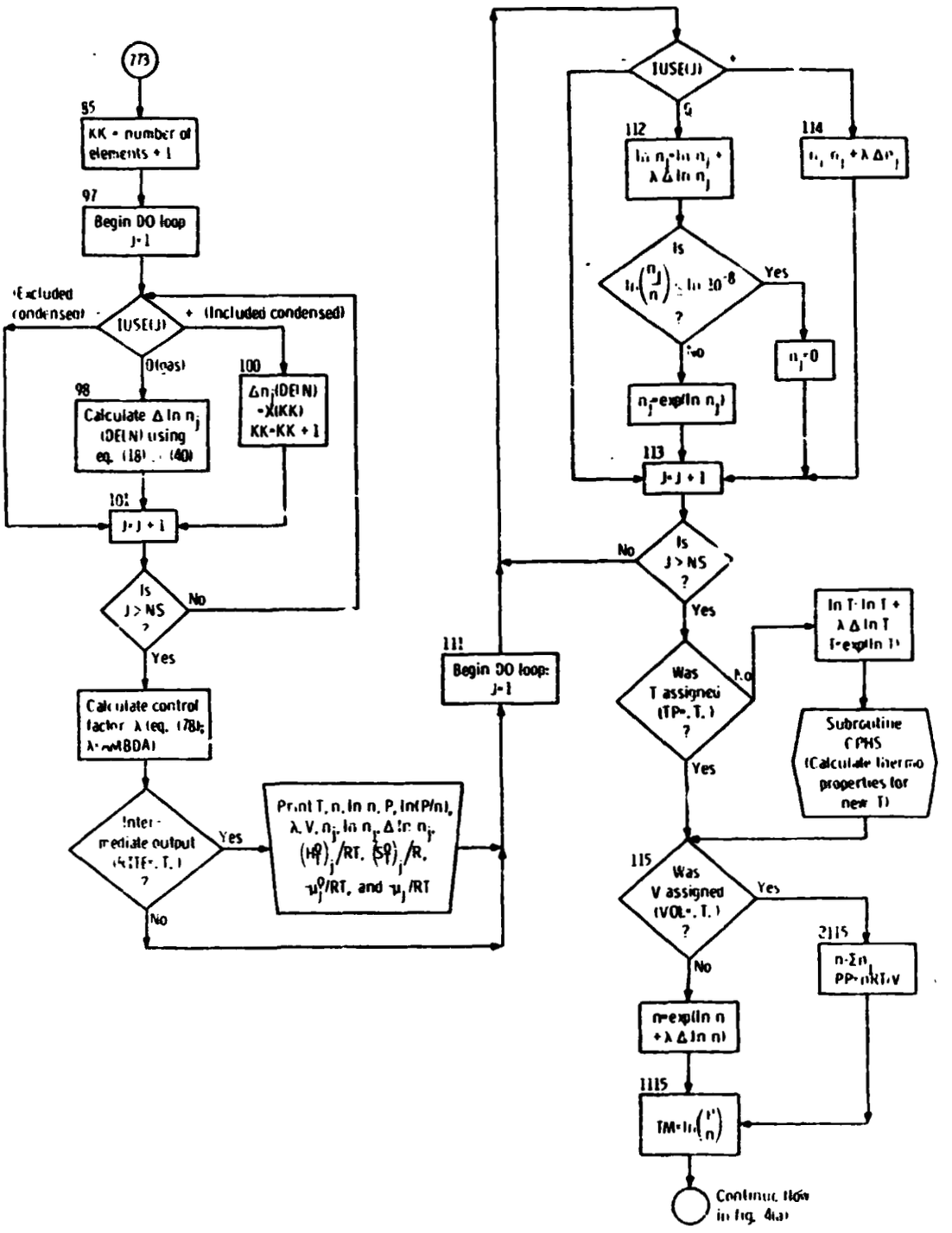


Fig. 5-1b Flow Diagram for Obtaining and Applying Corrections in Subroutine EQLBRM (Statement 85)

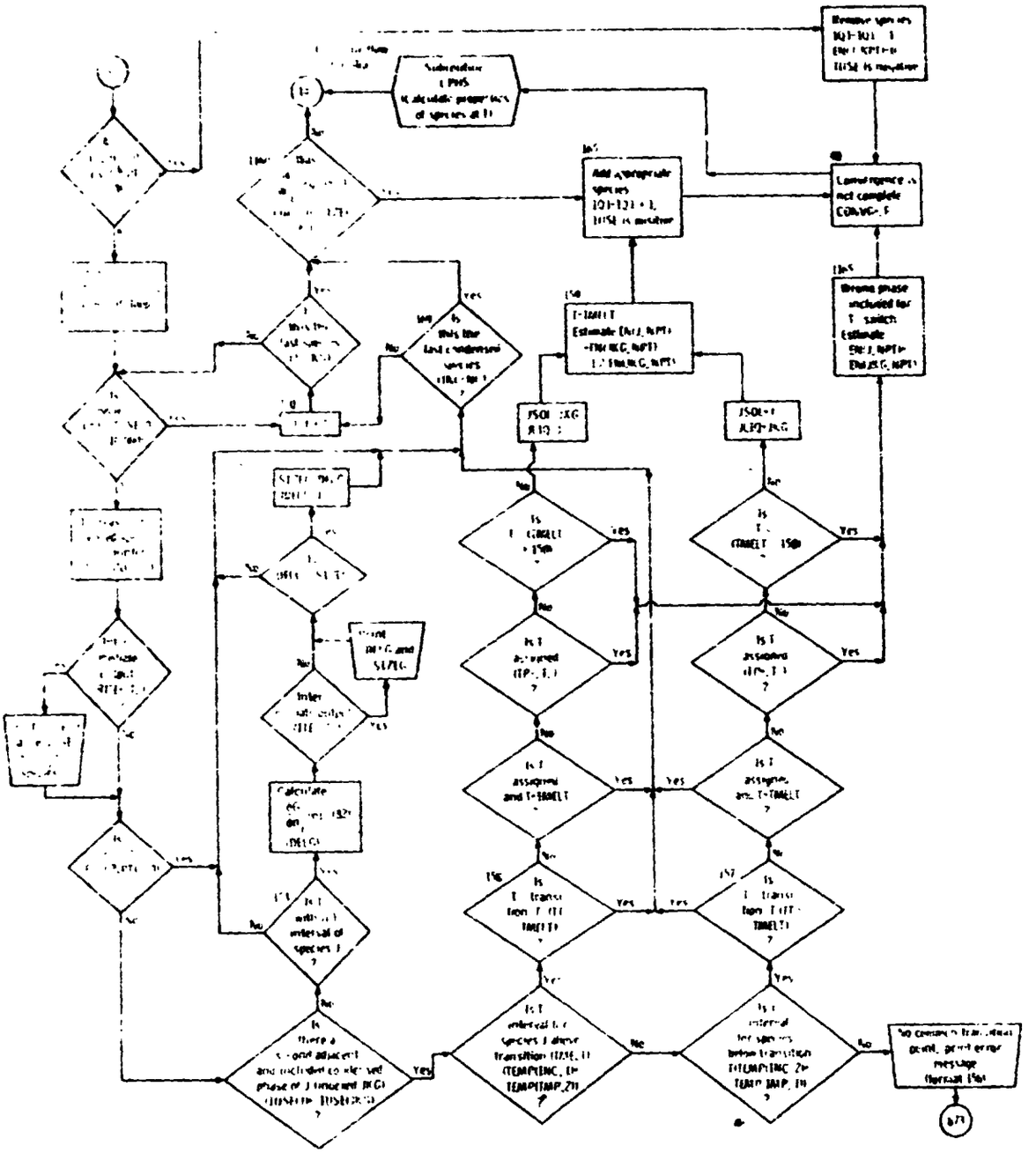


Fig. 5-1c - Flow Diagram for Adding and Removing Condensed Species in Subroutine EQLBRM

Common variables set by EQLBRM include TTT(NPT), PPP(NPT), SSUM(NPT), HSUM(NPT), CPR(NPT), GAMMAS(NPT), VLM(NPT), WM(NPT), DLVPT(NPT), DLVTP(NPT), TOTN(NPT), ENN, EN(j,NPT), ENLN(j), IUSE(j), JLIQ, and JSOL.

- Subroutine FROZEN

Subroutine FROZEN is called from ROCKET to calculate the temperature and thermodynamic properties for the following assigned conditions:

1. Composition frozen at combustion conditions (NFZ = 1)
2. An assigned exit pressure (PP)
3. An assigned entropy equal to the entropy at combustion conditions (SO = SSUM(1)).

If a temperature is reached 50K below the range of a condensed combustion species ((TEMP(j,1) to TEMP(j,2)), calculations are stopped. TT is set to zero and control is returned to ROCKET where a message is printed and data for all preceding points are listed.

The variables which must be set in common before calling FROZEN include NFZ, NPT, TT, PP, IUSE(j), COEF(k,i,j), SO, NS, NC, TEMP(j,k), TMID, WM(1), EN(j,1), RR, and TOTN(1). The variables which are set by FROZEN include TTT(NPT), PPP(NPT), SSUM(NPT), HSUM(NPT), CPR(NPT), GAMMAS(NPT), VLM(NPT), WM(NPT), DLVPT(NPT), DLVTP(NPT), and TOTN(NPT). Many of these variables are the same as are required by or set by EQLBRM.

- Subroutine GAUSS

Subroutine GAUSS is used to solve the set of simultaneous linear iteration equations constructed by subroutine MATRIX. The solution is effected by performing a Gauss reduction using a modified pivot technique. In this modified pivot technique only rows are interchanged. The row to be used

for the elimination of a variable is selected on the basis that the largest of its elements, after division by the leading element, must be smaller than the largest element of the other rows after division by their leading elements.

The solution vector is stored in X(k). In the event of a singularity, IMAT (which is equal to the number of rows) is set equal to IMAT - 1. IMAT is tested later in subroutine EQLBRM.

- Subroutine HCALC

The purpose of HCALC is to calculate thermodynamic properties for reactants under certain circumstances. HCALC is called from NEWOF (see "Entry NEWOF"), SHCK and DETON.

HCALC is called from NEWOF when CALCH is set true. CALCH Is set true in the main program when zeros have been punched in card columns 37 and 38 on one or more REACTANTS cards. The zeros are a code indicating that the enthalpy (or internal energy for UV problems) for the reactant should be calcualted from the THERMO data at the temperature punched on the card. This temperature has been stored in the RTEMP array. CPHS is called to calculate the enthalpy. The value is stored in the ENTH array and printed in the final tables.

The properties calculated in subroutine HCALC, their FORTRAN symbols, and the conditions for which they are used are as follows:

Property	FORTRAN Symbol	Equation	Conditions
$H(k)_T$	HPP(k)	(192)	SHOCK problem. DETN problem with T schedule. HP, RKT, or DETN problem if 00 in cc 37 and 38.
h,R	HSUBO	(193)	SHOCK problem. DETN problem with T schedule. HP, RKT, or DETN problem if 00 in cc 37 and 38.
$u(k)_T$	HPP(k)	(194)	UV problem if 00 in cc 37 and 38
u_o' R	HSUBO	(195)	UV problem if 009 in cc 37 and 38
M_o	AM	(197)	SHOCK or DETN problem
m_i	EN(j)	(205)	SHOCK problem
C_o	CPR1	(206)	SHOCK problem
S_o	SO	(207)	SHOCK problem

The quantity m_i was deliberately subscripted differently from EN(j) to allow for the fact that the same compound may have a different index as a reactant than as a reaction species. Thus, for example, $O_2(g)$ might be the third reactant read in from REACTANTS cards and also the tenth species read in by SEARCH. In this case m_3 would be stored in EN(10).

- Subroutine INPUT

Subroutine INPUT sets up the transport and relaxation data needed for the transport property calculations done in TRANSP. It is called from TRANSP for each point. The various functions of this subroutine are outlined as follows:

1. The EN array is searched for the most important gaseous species for the current point. These are identified and saved by storing the index of the species name in IND. A maximum of 20 species is allowed. All species of mole fractions less than 10^{-7} are omitted, as well as all condensed phases. However, all gaseous atomic elements are initially included, even if they are not among the 20 most important species, or even if their mole

fractions are less than 10^{-7}. This condition is imposed to satisfy a requirement imposed upon the A array, which is explained later in this section. If any elements have been omitted through use of an OMIT card, they are reinserted into the A array at this point. A message is printed out giving the name of the element reinserted into the A array: "NO ELEMENT WAS FOUND IN THE LIST OF SPECIES WITH THE NAME (name of species), OR ELSE THERE IS AN ERROR IN THE A(I,K) ARRAY."

2. The mole fractions and molecular weights are now calculated for the new reduced composition obtained in Step 1.

3. Transport and relaxation data are initialized to zero. Then, data stored in TABLES are searched, interaction by interaction, for data pertinent to the current point. When such data are found, subroutine LGRNGE(TT) is called. LGRNGE(TT) interpolates for the temperature TT. If data are missing for a pure species, an empirical equation is used to estimate the data. If data are missing for an interaction between unlike species, data are estimated from combining rules, using the data of the pure species. The empirical equation and the combining rules are described in the next section. If data for a pure species are missing, an error message is printed out: "NO TRANSPORT DATA WERE FOUND FOR THE SPECIES (name of species)." If the logical variable NODATA is not specified in the INPT2 namelist, the program sets NODATA = .F. and the message is printed. If NODATA = .T. is set in namelist INPT2, the message is not printed. However, no message is ever printed when data are missing for an interaction between unlike species. In either case, the program continues. The purpose of this error message is to warn the program user when transport data for a major species are missing. When the user is certain this is not the situation, he may wish to omit the message, in order to avoid getting the message every time the program fails to find data for a minor species.

4. The final operation of INPUT is to read the stoichiometric coefficients from the A array into the STC array and reorder them so as to express them as a set of chemical reaction equations. As was mentioned earlier, all the elements in the system are initialy included. This is not a necessary requirement, but was done as a matter of convenience. By including all the elements among the 20 (or less) gaseous species in the system, it is possible to use the A array to express the system in terms of a sufficient set of independent chemical equations. The number of required independent equations is given by taking the total number of species and subtracting the number of chemical elements in the system. So by choosing the set of chemical equations as the chemical reactions of formation of each species, a set of equations can be easily written directly from the stoichiometric coefficients in the A array. For instance, in the A array corresponding to the column for CH_4, there is a 1 in the row for carbon, a 4 in the row for hydrogen, and a 0 in the rows for the remaining elements. By assigning a -1 to CH_4, the chemical equation $C + 4H - CH_4 = 0$ is formed.

This procedure is applied to each species in the system, and the result is the initial set of equations. This initial set of equations is then reduced in order to eliminate any species or element (in this case always an element) not found among the 20 most important species in the system. This reduction is accomplished by searching through the chemical equations for an element with a mole fraction less than 10^{-7}, solving the chemical equation for that element, and then substituting the result in any other equation in which the element appears. The new set of equations is one less in number than the original set. This procedure is repeated until all the elements of mole fractions less than 10^{-7} have been eliminated from the chemical equations. The stoichiometric coefficients of this final set of equations are stored in STC and are used for calculating the reaction contribution to the heat capacity and thermal conductivity.

- Subroutine LGRNGE(TT)

Subroutine LGRNGE(TT) is a four-point Lagrange interpolation routine. It is used to interpolate within the tables of transport and relaxation data at temperature TT.

- Subroutine MASTER

MASTER is the main program of the TRAN72 program. It provides overall control of the operation of the program.

- Subroutine MAIN1

This subroutine determines how and where the input data should be read in. It provides overall control of the equilibrium thermodynamic property calculation.

- Subroutine MATRIX

This subroutine sets up the matrices corresponding to Tables I and IV. The assigned thermodynamic state being set up is specified by the following codes:

Assigned Thermodynamic State	Codes
TP	TP = .TRUE. VOL = .FALSE. CONVG = .FALSE.
HP	HP = .TRUE. VOL = .FALSE. CONVG = .FALSE.
SP	SP = .TRUE. VOL = .FALSE. CONVG = .FALSE.
TV	TP = .TRUE. VOL = .TRUE. CONVG = .FALSE.
UV	HP = .TRUE. VOL = .TRUE. CONVG = .FALSE.
SV	SP = .TRUE. VOL = .TRUE. CONVG = .FALSE.

After convergence of any of the previous six problems, setup of the derivative matrices is specified by the following codes:

Derivative	Codes
DLVTP	CONVG = .TRUE. LOGV = .FALSE.
DLVPT	CONVG = .TRUE. LOGV = .TRUE.

- Subroutine MOCDAT

This subroutine writes the results on Tape 10 in the particular format required by program PLUME.

- Subroutine OUT

The output routine, appropriately called OUT, handles all the output of the transport property calculations. This includes table headings, units, calculated data, spacing, and punched-card output.

This routine was written with the capability of saving the transport data for as many as 52 points. These data are saved in STORE. Including the current set of 13 points, this means that as many as 65 points can be printed at one time. If the problem has more than 65 points, which might occur for a TP problem, transport data will be printed out after every multiple of 65 points.

- Subroutine OUT1

This subroutine, together with entries OUT2 and OUT3, writes statements common to all problems. OUT1 writes statements giving the data on REACTANTS and for percent fuel, equivalence ratio, and density.

Entry OUT2. - This entry writes the statements for printing values of pressure, temperature, density, enthalpy, entropy, molecular weight, $(\partial \ln V/\partial \ln P)_T$ (if equilibrium). $(\partial \ln V/\partial \ln T)_p$ (if equilibrium), heat capacity, γ_S, and sonic velocity. These variables and corresponding labels are printed with a variable format described in BLOCK DATA.

Entry OUT3. - Entry OUT3 writes statements giving the equilibrium mole fractions of reaction species. It also cooperates in the removal of condensed species.

- Subroutine REACT

The purpose of subroutine REACT is to read and process the data on the REACTANTS cards. The subroutine is called from the main program after a REACTANTS code card has been read. The data on these cards are described in the Section 4.1 of Volume III.

The reactants may be divided into two groups according to card column 72 on the REACTANTS cards. The two groups are oxidants (0 in cc 72) and fuels (cc 72 $\neq$ 0). We generally keypunch F in card column 72 for fuels even though this is not necessary. The contents of card column 72 are read into FOX. Depending on the contents of FOX, program variables relating to oxidants or fuels are subscripted 1 for oxidants and 2 for fuels.

The FORTRAN symbols for the properties read from the REACTANTS cards and their associated properties discussed in INPUT CALCULTIONS (Section 4.1, Volume III) are as follows:

Property	FORTRAN symbol
$a_{ij}^{(k)}$	ANUM(j, m)[a]
$W_j^{(k)}$	PECWT(j) (if no M in cc 53)
$N_j^{(s)}$	PECWT(j) (if M in cc 53)
$\left(H_T^o\right)_j^{(k)}$	ENTH(j) (if not UV problem and 00 not in cc 37 and 38)
$\left(U_T^o\right)_j^{(s)}$	ENTH(j) (if UV problem and 00 not in cc 37 and 38)
$\rho_j^{(k)}$	DENS(j)

[a] Each of the j REACTANTS cards contains from 1 to 5 stoichiometric coefficients read (indicated by subscript m) into ANUM(j, m) and their corresponding chemical symbols read into NAME(i, m). In relating an ANUM(j, m) with $a_{ij}^{(k)}$, the index i associated with a particular chemical element is determined from the chemical symbol in NAME(j, m).

If there are several oxidants their properties are combined by subroutine REACT into properties of a total oxidant using the relative proportion of each oxidant given on the REACTANTS cards. Similarly, if there are several fuels, their properties are combined into properties of a total fuel. The total oxidant and total fuel properties discussed in INPUT CALCULATIONS and their associated FORTRAN symbols are as follows:

Property	FORTRAN symbol	Equation
$b_i^{(k)}$	BOP(i,k)	(187)
$M_j^{(k)}$	RMW(j)	(190)
$\mathcal{H}^{(k)}{}_T$	HPP(k) (if not UV problem and 00 not in cc 37 and 38)	(192)
$\mathcal{H}^{(k)}{}_T$	HPP(k) (if UV problem and 00 not in cc 37 and 38)	(194)
$M^{(k)}$	AM(k)	(196)
$\rho^{(k)}$	RH(k)	(198)
$V^{+(k)}$	VPLS(k)	(200)
$V^{-(k)}$	VMIN(k)	(201)
$w_j^{(k)} \Big/ \sum_{j=1}^{NREAC} w_j^{(k)}$	PECWT(j)	

If any of the $\rho_j^{(k)}$ are zero then RH(1) = RH(2) = 0.

These total oxidant and total fuel properties are subsequently combined into total reactant properties by using the values of oxidant-fuel mixture ratios obtained from the main program. This is done in NEWOF, an entry in SAVE.

Other common variables set by REACT are LLMT, NAME, ANUM, ENTH, FAZ. RTEMP, FOX, DENS, RMW, MOLES, NLM, NEWR, and NREAC.

A provision is made for eliminating a second tape search when two consecutive sets of REACTANTS cards contain the same elements. This is done by saving the element symbols (LLMT(ℓ)) in LLMTS(ℓ), the kilogram-atoms per kilogram (BOP(ℓ,k)) in SBOP(ℓ,k), and the number of elements (NLM) in NLS.

Atomic weights M_i are stored in ATOM(2,i). The corresponding chemical symbols are stored in ATOM(1,i). The oxidation states of the chemical elements V^+ or V^- are stored in ATOM(3,i). The ATOM array is stored by BLOCK DATA.

- Subroutine RKTOUT

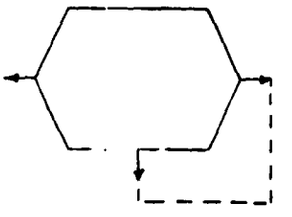
This subroutine calculates various rocket performance parameters from previoulsy calculated thermodynamic properties.

It is also the control program for writing rocket performance output. It contains the WRITE statements that apply specifically to rocket parameters and it calls subroutine OUT1 and entries OUT2 and OUT3 for the WRITE statements common to all problems. The rocket parameters are printed with the variable format, FMT, described in BLOCK DATA.

Subroutine RKTOUT is called from subroutine ROCKET.

- Subroutine ROCKET

This subroutine provides control for the rocket performance calculations. A logic-level flow chart is given in Fig. 5-1. The symbol is used to indicate that the loops shown are not actually coded as DO-loops. Note that there is a RETURN to CHEM and a reentry at ROCKT1 once in each loop.



Subroutine ROCKET obtains the required thermodynamic properties for equilibrium performance by calling subroutine EQLBRM. For frozen performance, subroutine ROCKET calls subroutine FROZEN to obtain the required thermodynamic properties. Rocket performance parameters are then obtained by calling subroutine RKTOUT. In addition to calling RKTOUT and FROZEN, subroutine ROCKET also:

- Calculates estimates for throat pressure ratios.
- Calculates estimates for pressure ratios corresponding to assigned area ratios (if any).

- Subroutine SAVE

Subroutine SAVE has several functions, all of which are concerned with saving some information from a completed calculation for subsequent use in later calculations. The primary purpose is to save computer time by having good initial estimates for compositions.

These estimates for the next point, NPT, come from either the point just completed, ISV, or some other previous point. The flow of the routine is directed by ISV as follows:

1. ISV positive. Transfer compositions for point just completed for use as initial estimates for next point (transfer EN(j,ISV) to EN(j,NPT)).

2. ISV negative. Save values of ENLN(j) for gases and EN(j) for condensed in SLN(j), ENN in ENSAVE, ENNL in ENLSAV, IQl in IQSAVE, JSOL in JSOLS, JLIQ in JLIQS, and NLM in LL1. (These values are saved because they are to be used as initial estimates for some future point and they may be overwritten in the meantime.) Make ISV positive and transfer EN(j,ISV) to EN(j,NPT).

3. ISV zero. Use the data previously saved (as discussed in (2)) as initial estimates for current point. Restore IUSE codes and inclusion or exclusion of "E" as an element for IONS option.

Entry NEWOF - NEWOF combined the properties of total oxidant and total fuel calculated in subroutine REACT with an O/F value to give properties for the total reactant. NEWOF is called for each mixture assigned in the MIX array in INPT2 namelist. It is called from either THERMP, ROCKET, SHCK, or DETON. The calculated properties and corresponding FORTRAN symbols are:

Property	FORTRAN Symbol	Equation
BO(i)	BO(i)	(191)
h_o/R	HSUBO (if not UV problem)	(193)
u_o/R	HSUBO (if UV problem)	(195)
ρ_o	RHOP	( )
r	EQRAT	

Subroutine HCALC is called by Entry NEWOF to calculate the enthalpies for each reactant that has zeros keypunched in card columns 37 and 38 in its REACTANTS card (see Table VI).

Values of HPP(2), HPP(1), HSUBO, BOP(i,2), BOP(i,1), and BO(i) are printed out.

- Subroutine SEARCH

Subroutine SEARCH reads the THERMO data which have been stored on tape 4 and stores the appropriate data in core.

A check is made near the beginning of the routine to prevent THERMO data from exceeding their storage allotments. These variables are all in labeled common SPECIES and are currently dimensioned for 150 species (Appendis B). However, this dimension may be reduced to save storage.

SEARCH is called from the main program when the logical variable NEWR is true. NEWR is set true in REACT to indicate a new chemical system. REACT also stores chemical element symbols for the current chemical system in the LLMT array. SEARCH stores THERMO data in core for each species whose elements are included in the LLMT array (unless the species name was listed on an OMIT card).

The THERMO data are stored in common variables TLOW, TMID, THIGH, SUB, A, COEF, and TEMP. SEARCH writes out the names ad dates of species whose data are stored ir core.

SEARCH initializes the IUSE array. IUSE(j) for gaseous species are set equal to zero. IUSE(j) for condensed species are set equal to negative integers. For the chemical system under consideration, the first possible condensed species is set equal to -1, the second to -2, and so on, with one exception. In the event there are two or more condensed phases of the same species, each phase is given the same negative integer. Thus, if IUSE(j) for $B_2O_3(\ell)$ is set equal to -4, for example, IUSE(j) for $B_2O_3(s)$ will also be set equal to -4. A description of the IUSE array is given in the next section.

The various condensed phases of a species are expected to be adjacent in the THERMO data as they are read from tape 4. These phases must be either in increasing or decreasing order according to their temperature intervals.

NS contains the total number of species stored in core. NC contains the total number of condensed species (counting each condensed phase of a species as a separate species).

IUSE array - Each value in the IUSE array is associated with a species. These values of IUSE serve two purposes:

1. They indicate which species are to be included in the current iteration (IUSE(j) $\leq$ 0 for excluded species and IUSE(j) $\geq$ 0 for included species).

2. They indicate multiple phases of the same species if abolute values of IUSE(j) are equal.

The IUSE(j) are initialized in subroutine SEARCH and the main program as follows:

1. IUSE(j) = 0 for all gaseous species.

2. IUSE(j) = n for all condensed species whose names have been listed on INSERT cards. The number n indicates the species was the n^{th} condensed species whose THERMO data were read from tape 4.

3. IUSE(j) = -n for all condensed species not listed on INSERT cards where n is defined in (2).

These initial values of IUSE(j) may be adjusted later in subroutine EQLBRM. For condensed species, the sign is adjusted as species are included or excluded in the current iteration.

For the IONS option, IUSE(j) values for ionic species are set to -10000 when the mole fractions of all ionic species are less than 10^{-8}.

- Subroutine SET

This routine initializes the common arrays which contain concentration information to zero and provides an initial guess (ENN = 0.1) for the inverse of the mixture molecular weight and also for the chamber temperature (TT = 3800 K). The IUSE array is also initialized.

- Subroutine TRANSP

Subroutine TRANSP is the main routine for the transport calculations. A flow diagram is given in Fig. 5-1. All calculations of the properties are done in this routine. The equations are given in the next section.

One other operation is carried out in this routine. Local tape unit 3 is searched for the transport and relaxation data of the important interactions and saved in the variable TABLES. This search differs from the one in subroutine SEARCH. In SEARCH, data are saved for all interactions in the chemical system; whereas, the TRANSP, interactions involving a trace species are eliminated.

The remainder of the routine is the calculation of the properties. Calculation of the viscosity, monatomic thermal conductivity, reaction thermal conductivity, and reaction heat capacity all involve the solving of a set of simultaneous linear equations. The matrix elements for each are calculated in TRANSP, but the actual solution is obtained from subroutine GAUSS.

The solutions obtained from GAUSS are checked for accuracy for two of the properties, viscosity and reaction thermal conductivity. If the initial equations are not satisfied to a prescribed tolerance by using the solution obtained from GAUSS, an error message is printed out.

- Subroutine VARFMT

Subroutine VARFMT (variable format) adjusts the number of decimal places printed in F-format in the variable format, FMT, according to the size of the number. It is used for P_c/P_e, P, and A_e/A_t. Variable format is described in BLOCK DATA.

5.3 DESCRIPTION OF BLIMPJ SUBROUTINES

The BLIMP-J subroutines are identified by two labels. The first label is the element name and the second label is the subroutine name, e.g., B03A (element name), SETUP (subroutine name). In the following description the subroutines are ordered according to their element name. Figure 5-2 gives a flow chart which shows the general solution procedure and the interconnection of the major subroutines.

There are several dummy subroutines included in the program. Some of these are for obtaining information from the computer system, e.g., date, time of day, etc. The specific routines are B30B, B30D, B30E, B30F, B30G. They are described on the following pages. If there are system subroutines of the same name and function they may be removed from the program. Alternatively, they may be used to call the appropriate system routine.

- B01A DUMCOM

A collection of all labeled commons sometimes useful when performing debug operations. (Serves as main program for CDC machines. Calls BLIMP (B02A).

- B02A BLIMP

Master calling program. For the Univac system, this program calls SETUP, ITERAT, OUTPUT, ROCOUT. (For CDC this is a subroutine called by DUMCOM).

- B03A SETUP

Control program for setting up boundary layer edge conditions and streamwise derivatives for a new station or a new case. Called by BLIMP. Calls FIRSTG, LINMAT, RECASE, TRMBL, STATEN, REFCON, TRANCR, HISTXI, INPUT, TOD, ETIMEF, DATE.

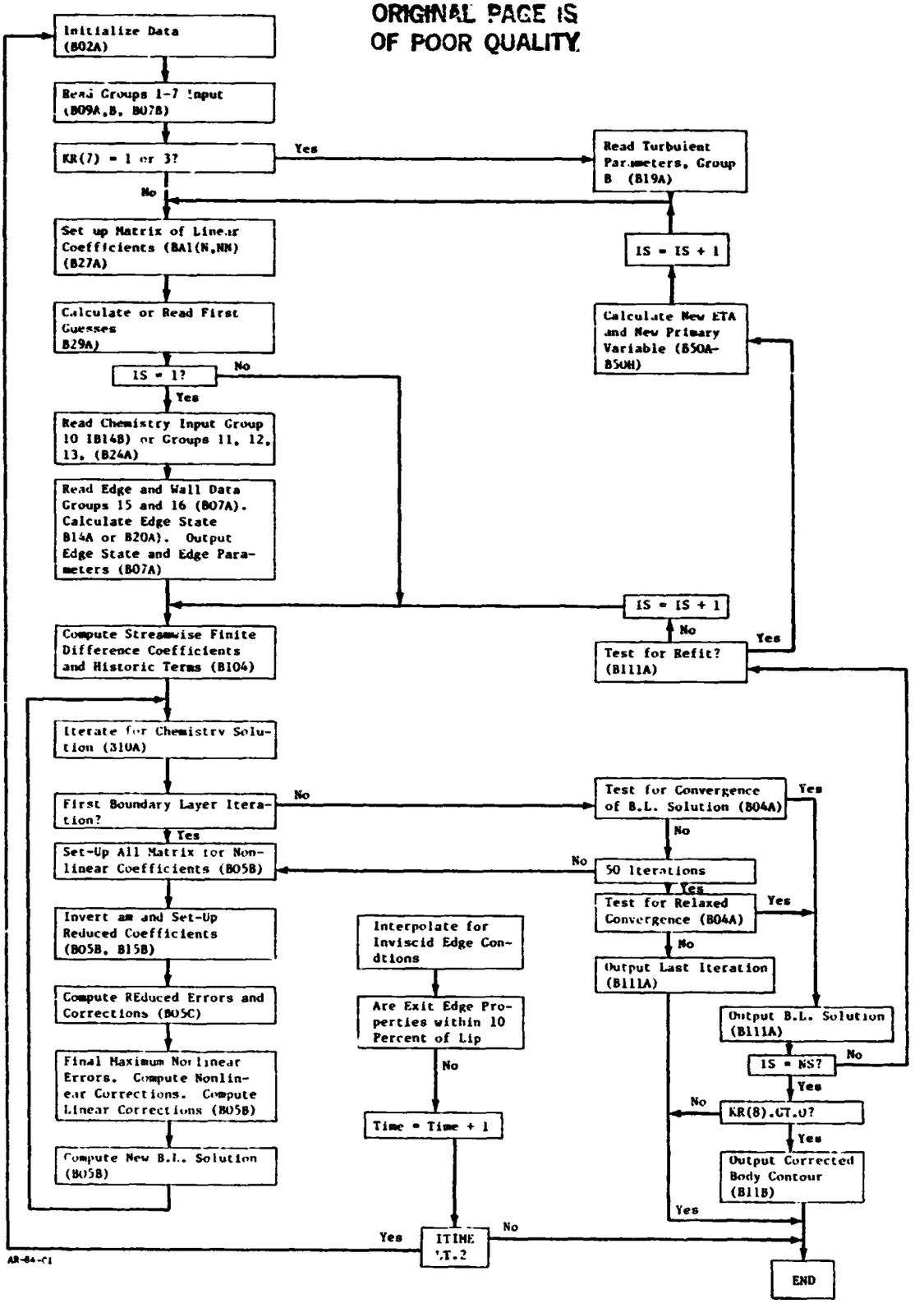


Fig. 5-2 Flow Chart for BLIMP-J Solution Procedure

- B04A ITERAT

Control program for performing boundary layer interaction and testing maximum errors for convergence. Called by BLIMP. Calls NNNCER, ETIMEF, NONCER, TLEFT.

- B05B NNNCER (entry point NONCER)

Control program for performing that portion of a boundary layer iteration having to do with solution of the nonlinear (conservation) equations. With the aid of its subroutines, it evaluates errors and coefficients of the corrections of the nonlinear equations, reduces this matrix to the nonlinear set, evaluates maximum errors of conservation equations, evaluates corrections, computes damping factor and applies to corrections, and corrects primary variables. Called by ITERAT. Calls IMONE, EQUIL, ICOFF, IONLY, RERAY, ABMAX, RNLCER, STATE, OGLE, LINCER, TRMBL, TRANCR, LIAD, ETIMEF.

- B05A RNLCER

Further reduces nonlinear equations to reduced nonlinear set of wall variables. Introduces wall boundary conditions and solves for new values of this set. Called by NNNCER. Calls RERAY, EQUIL.

- B06A LINCER

Evaluates errors for linear equations (i.e., Taylor series expansions and linear boundary conditions) and with the aid of its subroutines, determines maximum errors of linear equations and corrects errors for these linear equations for the matrix reduction which is performed on the linea equations (see discussion under subroutine MATS1). Called by NNNCER. Calls ABMAX, MATS1, MATS2.

- B07A REFCON

Calculates boundary layer edge conditions and sets up wall boundary conditions for uncoupled problems. Called by SETUP. Calls STATE, EQUIL, SLOPQ, SLOPL.

- B07B MISCIN

Sets up default values for certain variables and reads namelist $MISLIS. Called by RECASE.

- B08B ICOEFF

Calculates groupings which contribute to the error equations and influence coefficients for the nonlinear (conservation) equations. Called by NNNCER.

- B09A RECASE

Reads in most of boundary layer input data Called by SETUP. Calls TOD, DATE, GEOM.

- B09B GEOM (S, R, P, KIN, NBT, NBT2, NS, PTET, NTH, GE, IP, IU)

Reads namelist $INPUT and computes the wall length and the gradients of pressure and velocity when necessary. Selects from the input data those stations used for boundary layer solution stations. Called by RECASE.

S - wall length
R - nozzle radius
P - pressure
KIN,NBT,NBT2 - unit assignments
NS - number of BLIMP solution stations

PTET - axial coordinates of BLIMP solution stations

NTH - throat station number

GE - $\cos\phi$ (wall angle)

IP - flag for input of edge pressure and edge pressure gradient

IU - flag for input of edge velocity and edge velocity gradient.

$$s_i \text{ (wall length)} = s_i + \sum_{j=2}^{NP_i} \left[(r_j - r_{j-1})^2 + (x_j - x_{j-1})^2\right]^{1/2}$$

$$\phi_i = \arctan\left(\frac{r_j - r_{j-1}}{x_j - x_{j-1}}\right), \quad j = NP_i$$

for IP = 1

$$\left.\frac{dP}{dx}\right|_i = \frac{1}{2}\left[\frac{P_{j+1} - P_j}{x_{j+1} - x_j} + \frac{P_j - P_{j-1}}{x_j - x_{j-1}}\right], \quad j = NP_i$$

for IU = 1

$$\left.\frac{dU_e}{dx}\right|_i \quad \text{same as} \quad \left.\frac{dP}{dx}\right|_i \quad \text{with P replaced by } U_e$$

where i - BLIMP solution station

j - index on input x, r, P, etc.

NP_i - value of j for the i^{th} BLIMP solution station.

- B10A

Computes terms involving derivatives with respect to XI (i.e., nonsimilar terms) and stores those upstream quantities needed for these difference relations. Called by SETUP. Calls TAYLOR.

- B11A OUTPUT

Prints standard boundary layer output block for converged solution or, if required, at the end of each iteration. Called for BLIMP. Calls REFIT.

- B11B ROCOUT

Available as an option (KR(8) = 1,2,3), this subroutine calculates a corrected body contour which can be output onto punched cards for use as input to TDK.

The KR(8) = 1 option calculates and punches the inviscid flow contour which should be used for TDK input for a specified, and different, nozzle contour (which has been input to BLIMP-J). The inviscid contou· calculated from

$$R_I = R_B - \delta_B^* \cos o$$

$$X_I = X_B = \delta_B^* \sin o$$

where R_I is the inviscid contour radius, R_B is the nozzle radius (input), δ_B^* is the body displacement thickness, and ϕ is the wall angle.

The KR(8) = 2 option calculates and punches the desired body contour if the input contour is the inviscid flow field contour. The body contour is calculated from

$$R_B = R_I + \delta_B^* \cos\phi$$

$$X_B = X_I - \delta_B^* \sin\phi$$

where the terms are the same as above except that R_I is the input contour to BLIMP.

In both cases the contour is normalized to the throat radius (the minimum radius) and the axial coordinate is zero at the throat. Also, the contour is punched in a form suitable for TDK input.

- B12B IMONE

Evaluates the coefficients of the $(I-1)^{th}$ corrections for the I nonlinear (conservation) equations, where I is the I^{th} nodal point in the boundary layer. Called by NNCER. Calls TAYLOR, LIAD.

- B13B IONLY

Evaluates the coefficients of the I^{th} corrections for the I^{th} nonlinear (conservation) equations, where I is the I^{th} nodal point in the boundary layer. Called by NNNCER. Calls LIAD.

- B14A STATE

Evaluates the chemical state and properties of a homogeneous gas mixture. Called by NNNCER, REFCON. Calls HHOMO, CHOMO, SHOMO.

- B14B STATEN

Reads in basic property data for homogeneous boundary-layer option. Called by SETUP.

- B14C HHOMO(T)

Calculates enthalpy of homogeneous gas at temperature T, degrees R. Called by STATE.

- B14D CHOMO(T)

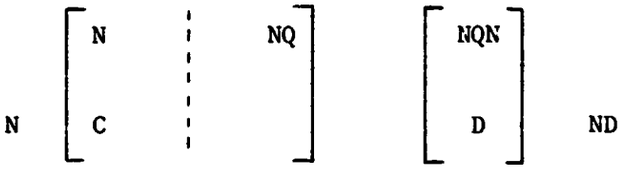
Calculates specific heat of homogeneous gas at temperature T, degrees R. Called by STATE.

- B14E SHOMO(T)

Calculates entropy of homogeneous gas at temperature T, degrees R. Called by STATE.

- B15B RERAY (N, C, NQ, D, NQN, NNN, LS, IS, ND, SD, L, S, LL, LLL)

Replaces rectangular matrix (C) with N rows of N+NQ columns by the product of the inverse of an N by N submatrix and the remaining columns of C. The inverse is also permitted to act on additional columns (matrix (D) with ND rows and NQN columns) from another portion of memory. Also. routine rearranges columns according to arbitrary specifications given by LS.



Called by EQUIL, NNNCER, RNLCER.

N = number of rows in rectangular matrix (see sketch)
C = elements of rectangular matrix (see sketch)
NQ = number of columns in matrix C in excess of those contributing to square matrix (see sketch)
D = elements of matrix of additional columns (see sketch)
NQN = number of additional columns (see sketch)
LS = sequence to which columns of C are rearranged (LS(1) = 0 signifies no rearrangement)
IS = flag, yields debug output if RERAY entered with IS = -2, signifies singular matrix if RERAY yields IS less than zero
ND = dimension on rows of C from calling program SD-LLL used to bring in dummy storage space.

- B16A SLOPQ (N, X, Y, S, Z)

Based on a sequence of quadratic (3-point) fits of a set of points, calculates average slope at each point and integrates the equation thus defined between each pari of points. Called by REFCON.

N = number of points to be considered
X = abscissa at each point
Y = ordinate at each point
S = derivative at each point
Z = integral up to each point.

- B16B SLOPL (N, X, Y, S, Z)

This routine performs the same function as B16A SLOPQ except tha linear (2-point) fits are used instead of quadratic (3-point) fits. The slope is the average of the left and right slopes. Called by REFCON, TRANCR.

- B17A ABMAX (N, X, XM, I)

Searches an array for the entry with maximum value. Called by LINCER, NNNCER.

N = number of entries in the array
X = coefficients in array under consideration
XM = entry with maximum absolute value
I = index on XM.

- B18A MATS1(X)

Performs operations on a column of a matrix B or on a column of errors R (designated X in call list) such as to form A**(-1)*X where A**(-1) is the

inverse of the sparce matrix formed from the Taylor series expansions of F(1,I)) and their derivatives (in the case of MATS1) and of G(1,I) or SP(1,I,K), vix.,

Original matrix operation

$$(A + B)V = R$$

multiplying through A**(-1)

$$[1 + A^{**}(-1)*B]V = A^{**}(-1)*R$$

Called by LINCER, LINMAT, MATS2.

- B18B MATS2(X)

  See MATS1 for function. Called by LINCER, LINMAT, FIRSTG. Calls MATS1.

- B19A TRMBL(ILK)

  Evaluates turbulent transport properties and their derivatives with respect to nonlinear variables. Called by SETUP, NNCER. Calls LIAD, TAYLOR, ERP, ERF.

- B19B ERF(X)

  Calculates the error function of X. Called by TRMBL.

- B19T TRANCR

  Evaluates terms required for consideration of transverse curvature. Called by SETUP, NNNCER.

- B20A EQUIL (KQ, Z, PRR)

Control program for computation of chemical state of the system. Performs such complex functions as setting up for different types of solutions (isentropic expansion, stagnation point, boundary layer or wall), recalling stored values of boundary layer solutions and reinitializing omitted species, re-evaluating absent atom array, deleting molecules based on absent atom array, and, with the aid of subroutines, evaluating properties, controlling principal iterative loop, and reinverting and attempting alternate paths when convergence problems occur. Called by NNNCER, REFCON, RNCLER. Calls CRECT, MATER, PROPS, RERAY, THERM.

KQ = flag which controls chemistry options (see FORTRAN variables list)
Z = enthalpy (when used)
PRR = pressure.

- B21A TERM

Evaluates current thermodynamic properties for each species, which data are required for evaluation of errors and correction coefficients in chemistry solution. Called by EQUIL.

- B22A MATER

Evaluates current errors in chemistry solution and sets up matrix of linearized correction equations. Called by EQUIL. Calls KINET.

- B23A CRECT(MOE)

Corrects state variables and composition, principal logic being involved with limiting corrections such that instabilities in the iterations will not occur. Called by EQUIL.

MOE = 0 or 1 if linearization done predominantly on equilibrium or mass balance relations, respectively.

- B24A INPUT

Reads in basic elemental composition data and species property data, selects base species, and sets up stoichiometric coefficients for species formation reactions. Called by SETUP.

- B25A PROPS

Computes all properties and property derivatives required by boundary layer calculations. Called by EQUIL.

- B26A TAYLOR (D, FM, F, P)

Calculates coefficients in Taylor series expansions of integrals which appear in the integral form of the boundary layer equations. Called by HISTXI, AMONE, TRM3L.

D = distance between neighboring nodes I and I-1
FM = value of function and its derivatives at I-1
F = value of function and its derivatives at I
P = terms in Taylor series expansion.

- B27A LINMAT

Sets up matrices for Taylor series expansions and linear boundary conditions from eta spacing, and solves to express linear corrections in terms of nonlinear corrections. Called by SETUP. Calls MATS1, MATS2.

- B28A KINET

The subroutine is reserved for modeling of kinetically controlled surface reactions. called by MATER.

- B29A FIRSTG

Computes or reads in first guesses for primary variables or instructs program to use values from previous case. Called by SETUP. Calls MATS2, MATS1.

- B30A ERP(X)

Forms Dawson integral of X. Called by TRMBL.

- B30B ETIMEF(T) (entry point ETIME)

Subroutine to call the system for elapsed time, T, in seconds. Present routine calls the system by a call SECOND. This call should be replaced with the appropriate system call, or the entire subroutine can be replaced by a dummy. Called be SETUP, ITERAT, NNNCER.

- B30C LIAD (L, I, J, C)

Alters elements of the AM matrix and the corresponding errors to reflect the solutions to the linear equations. Called by NNNCER, IMONE, IONLY, TRMBL.

L = 1 for momentum, 0 for enthalpy, and K for species equations
I = K^{th} nonlinear equation
J = J^{th} linear variable
C = coefficient of j^{th} linear variable in I^{th} nonlinear equation

- B30D TLEFT(I)

Dummy subroutine. Not used with BLIMP-J. Called by ITERAT.

- B30E DATA (I, J)

Dummy subroutine. Can be replaced with a call to the system for date. Called by SETUP, RECASE.

I = 9
J is dimensioned 3 and is expecting a format of 3A6. The first 9 locations are filled by DATE and the second 9 locations by TOD.

- B30F TOD (I, J)

Dummy subroutine. Can be replaced with a call to the system for time of day. Called by SETUP, RECASE.

I = 18
J = see B30E

This subroutine and B30E fill the J(3) with information giving date and time of day.

Example: 10 AUG 74 10:23:02

- B30G SECOND(T)

Dummy subroutine. Called by ETIMEF.

- B36A OGLE (N, XAM, PRM, DPDIM, NUMX, X, P, EM)

Looks up an array of values of a single dependent variable using a cubic curve fit between any two points (and corresponding two slopes) of the table. Called by NNMCER.

N = number of points to be considered
SAM = value of independent variable for which lookup is to be performed
PRM = output interpolated values returned by OGLE
DPDIM = output interpolated slopes returned by OGLE
NUMX = numver of tabular entries in the table
X = tabular independent variable
P = tabular dependent variable
EM = slopes to be used.

- B50A FILQ3

This routine converts the coordinate and constraint data into elements in the solution matrix and sets up this matrix for FINEQ. Calls FUNXS, TRINT.

- B50B FILQ5

This routine evaluates values of variables and their derivatives at new nodes. Called by FISLEQ. Calls FUNXS.

- B50C FINEQ

This routine solves for the unknown coefficients of the new polynomial segments based on LU matrix decomposition. Called by FISLEQ.

- B50D FISLEQ

This is the main subroutine for least square curve fits of variables between nodal points. Called by POINTS. Calls FILQ3, FINEQ, FILQ5.

- B50E FUNXS

This routine evaluates special polynomials for the refitting function. Called by FILQ3, FILQ5.

- B50F TRINT

This routine evaluates special polynomials for the refitting function. Called by FILQ3.

- B50G POINTS

This routine uses current values of the variables and tneir derivatives and solves for the coefficients of the polynomial segments between each pair of adjacent nodes. Limits placed on the velocity variable establish the new nodal distribution and values of remaining variables and their derivatives are calculated for this new distribution. Called by REFIT. Calls FISLEQ.

- B50H REFIT

This is the main calling routine for the refit procedure. It evaluates certain constraints which depend on NETA and the type of curve fit. The B50 subroutines ar all part of the REFIT option. Called by OUTPUT. Calls POINTS.

- EDGPRP EDGPRP

This routine determines the static pressure at the edge of the boundary layer in the inviscid RAMP nozzle flowfield after the nozzle boundry layer has been calculated. The first pass through the boundary layer solution uses static conditions at the nozzle wall as edge conditions in the boundary layer solution. If the static pressure at the nozzle wall are met within 10 percent of the pressure at the edge of the boundry layer as determined by EDGPRP then the boundary layer solution is run with edge conditions determined by EDGPRP in the inviscid flowfield at the edge of the boundary layer.

- EDGHSP EDGHSP

Subroutine EDGHSP determines total conditions of the boundary layer edge at the exit plane of a two phase nozzle. These total conditions can be used to rerun the boundary layer solution to better match static properties at the edge of the boundary layer in two phase nozzles.

There are severe gradients in total conditions near the lip of the nozzle for solid rocket motors. These gradients are caused by two phase effects and the rapid change in concentration of particles near the lip. The first pass through the boundary layer solution uses total conditions which match static properties at the nozzle lip. However, conditions at the boundary layer edge (in the inviscid flowfield) may not be matched. The edge total conditions calculated by this routine provide a better match of all static conditions at the boundary layer edge so that another pass through the boundary layer solution can be made to better simulate the edge conditions.

- IDMTAD IDMTAD

This function computes the gas property storage locations within the TABB array. This function is used by subroutine EDGHSP to determine boundary layer edge total properties from the TRAN72 equilibrium data tables transferred to the BLIMPJ code on the RAMP output data tape.

- READP READP

This subroutine reads the flowfield data from the RAMP nozzle flowfield output file (Unit 3). These data are used by Subroutine EDGPRP to establish boundary layer edge conditions for the second pass through the boundary layer solution.

- [illegible]AD

The function is used to locate a particular equilibrium flow property within the FABB array based on indices in the calling sequence. This routine is used to simulate a four dimensional array.

- TAPMOV TAPMOV

This routine is used to space past information in the RAMP output tape to get to the beginning of the RAMP computed flowfield data.

6. REFERENCES

1. Smith, S.D., "High Altitude Supersonic Flow of Chemically Reacting Gas-Particle Mixtures - Volume I - A Theoretical Analysis and Development of the Numerical Solution," LMSC-HREC TR D867400-I.

2. Smith, S.D., "High Altitude Supersonic Flow of Chemically Reacting Gas-Particle Mixtures - Volume III - RAMP2 - Computer Code Users and Applications Manual," LMSC-HREC TR D867400-III.

3. Svehlas, R.A., and B.J. McBride, "FORTRAN IV Computer Program for Calculation of Thermodynamics and Transport Properties of Complex Chemical Systems," NASA TN D-7056, January 1976.

4. Evans, R.M., "Boundary Layer Integral Matrix Procedure BLIMP-J User's Manual," Aerotherm UM-75-64, July 1975.

5. Bird, G.A., "Breakdown of Translational and Rotational Equilibrium in Gaseous Expansions," AIAA J., Vol. 8, November 1970, pp. 1998-2003.

6. Hendricks, W.L. et al., "Laser and Mixing Program (LAMP) Theory and User's Guide," TR R4-CR-77-4, Lockheed Missiles & Space Company, Huntsville, Ala., August 1976.

7. Nickerson, G.R. et al., "A Computer Program for the Prediction of Solid Propellant Rocket Motor Performance, Vol. II, AFPRL TR-80-34, Software and Engineering Assoc., Santa Ana, Calif., April 1981.

8. Kliegel, J.R. and J.N. Levine, "Transonic Flow in Small Throat Radius of Curvature Nozzles," AIAA J., Vol. 7, NO. 7, July 1969, pp. 1375-1378.

9. Kliegel, J.R., and G.R. Nickerson, "Axisymmetric Two-Phase Perfect Gas Performance Program - Vol. I," NASA CR 92069, April 1967.

10. Crowe, C.J., "Drag Coefficient of Particles in a Rocket Nozzle," AIAA J., Vol. 5, No. 5, May 1967.

11. Hall, I.M., "Transonic Flow in Two-Dimensional and Axially-Symmetric Nozzles," Quarterly J. of Mechanics & Applied Mechanics, Vol. XV, Pt. 4, 1962, pp. 487-508.

12. Smith, S.D. and A.W. Ratliff, "Rocket Exhaust Plume Computer Program Improvement - Vol. IV," LMSC-HREC D162220-IV-A, Lockheed Missiles & Space Company, Huntsville, Ala., January 1972.

13. Prozan, R.J., "Solution of Non-Isoenergetic Supersonic Flows by Method of Characteristics - Vol. III," LMSC-HREC D162220-III, Lockheed Missiles & Space Company, Huntville, Ala., July 1971.

14. Stephens, J.T., and A.W. Ratliff, "Studies of Rocket Engine Combustion Chamber Geometry Using an Equilibrium Reacting Gas Transonic Flow Program," LMSC-HREC A784898, Lockheed Missiles & Space Company, Huntsville, Ala., Nov. 1967.

Appendix A

GUIDE FOR CONVERTING THE RAMP2 PROGRAM TO THE CDC 7600

Appendix A

The RAMP2 program is operational on the Univac 1108 Exec 8 and CDC 7600 FTN4.8 systems. The known differences between the Univac and CDC systems are:

- The inclusion of PROGRAM cards in the main routines of the three programs.

  TRAN72 (MASTER)
  PROGRAM MASTER (INPUT, OUTPUT, PUNCH, TAPE5=INPUT, TAPE6=OUTPUT,
  *TAPE7=PUNCH, TAPE4, TAPE8, TAPE10, TAPE9, TAPE3).

  RAMP2F (MAIN)
  PROGRAM MAIN (INPUT, OUTPUT, PUNCH, TAPE5=INPUT, TAPE6=OUTPUT,
  *TAPE7=PUNCH, TAPE1, TAPE2, TAPE3, TAPE4, TAPE8, TAPE9, TAPE10,
  *TAPE11, TAPE12, TAPE13).

  BLIMPJ(DUMCOM)
  Program DUMCOM (INPUT, OUTPUT, PUNCH, TAPE5=INPUT, TAPE6=OUTPUT,
  *TAPE7=PUNCH, TAPE1, TAPE2, TAPE3, TAPE4, TAPE10, TAPE18, TAPE19,
  *TAPE20).

- All routines in the RAMP2F code which have the particle array PFPARY as an argument in the calling sequence and the MAIN program require a LEVEL 2, PFPARY statement so that for the CDC 7600 this array may be stored as Large Core Memory (LCM).

The Univac 1108 Exec 8 overlay instructions for the RAMP2F and BLIMPJ programs are shown in Figs. A-1 and A-2, respectively. Table A-3 and A-4 present the CDC 7600 segment loader instructions for the RAMP2F and BLIMPJ codes. In order for the CDC 7600 loader to operate the following dummy routines must be included for the RAM2F: LEGOO, LKB1, LKB2, LKB3, LKB4, LKB5, LKC1, LKC2, LKC3, LKC4, LKC5. The BLIMPJ program requires the following dummy routines: LEGOO, LINKA, LINK1, LINK2, LINK3, LINKY, LINK5, LINK6 and LINK7.

The TRAN72 does not require overlaying.

```
SEG MAIN
     IN TPF$.MAIN,.DRIVER,.BLKDAT
SEG B1*,(MAIN)
     IN TPF$.IDMTAB,.PLUMIN,.AXIS,.IDMPHI,.LOGIC,.POINT,.WG,.WTT
     IN TPF$.WXANDR,.MASS,.MAXT,.OFSET,.OUTPUT,.PAGVOF,.PHI,.STARTV
     IN TPF$.THERMT,.INITP,.GASRD,.IDMXSI,.IDTAPE,.GASTAP,.IMPUT
     IN TPF$.PARTIN,.LIPIN,.WOFA,.AOASTR,.SETHTG,.PARTPH,.MASCON
     IN TPF$.PLMOUT,.STLINE,.SPACET
SEG B2*,(MAIN)
     IN TPF$.TRANS,.COEFS,.DLTA,.FIXIL,.HALL,.ONED,.ORTHLS,.PARTIL
     IN TPF$.TRACE,.WDGI,.PROP,.PUNEX,.ARASSL,.SPECIE,.IDMPOP,.POP
SEG B3*,(MAIN)
     IN TPF$.BLMPIN,.IDMTOB,.SPECIB,.WALPRP,.TCB
     IN TPF$.SPECC,.IDSPEC
SEG B4*,(MAIN)
     IN TPF$.PRLTRC,.IDMPRO,.PRO,.IDMPDT,.PDT,.BLEXIT,.PARLOK
     IN TPF$.INRSCP,.TAPMOV,.READF,.INTERP,.TRACEP,.GAS,.START
     IN TPF$.PREAD,.WTFLOP,.WRITP
SEG B5*,(MAIN)
     IN TPF$.CHECK,.CHEM,.SLOP,.COEFEQ,.COTPRD,.EMOFP,.FNEWTN,.GAPPBI
     IN TPF$.INRSCT,.NEWENT,.PPATPT,.RGMOFP,.VEMAG,.VMODEL,.VMOD1
     IN TPF$.PHASE1,.SOKINT
SEG C1*,(B5)
     IN TPF$.STRNOP,.SOKSOL,.PHYSOL,.CARCTR,.SLPLIN
     IN TPF$.AVERAG,.STGMOD
SEG C2*,(B5)
     IN TPF$.NORSCK,.POFH,.VMOD2,.TURN,.OUT,.ESHOCK,.WEAK
     IN TPF$.DELTAF,.ENTROP,.LIMITS,.SLSKIP,.PRFRBD
     IN TPF$.FREEMC,.WTFLOF,.ITARM
     IN TPF$.TOFENH,.OUTBIN,.INTT
SEG C3*,(B5)
     IN TPF$.MASSCK,.INTEGR,.CBREAK,.THRUST,.ITERM
SEG C4*,(B5)
     IN TPF$.MCCRMK,.BACWRD,.CODEE,.CODEF,.CODEH,.DECODE,.FORWRD
     * TPF$.BOUNDA
SEG C5*,(B5)
     IN TPF$.THETPM,.PRANDT,.HYPER,.OVERFX,.MOCSOL,.EXPCOR
     IN TPF$.IRAD,.PLOAD,.NORMCK,.SLINT,.PARINT,.PARSTR
     IN TPF$.PAFH,.MESH,.WEKK,.ENTRPP,.DFLTFF,.ESHOCC
SEG D1*,()
     IN DUMSYS,PARTP1
END
```

Fig. A-1 RAMP2F Loader Instructions for the Univac 1108, EXEC 8 Operating Ststem

```
SEG ONE
IN B02A
SEG TWO*,(ONE)
IN B03A
SEG THREE*,(TWO)
IN B07A
SEG FOUR*,(THO)
IN B05A
SEG FIVE*,(TWO)
IN B24A
SEG SIX*,(TWO)
IN B14F
SEG SEVEN*,(ONE)
IN B04A
SEG EIGHT*,(SEVEN)
IN B11A
SEG NINE*,(SEVEN)
IN B11B
SEG TEN*,(ONE)
IN TPFS,EDGPRP,,EDGHSP,,TAPMOV,,READP,,IOMTAD,,TAD
```

Fig. A-2 BLIMPJ Loader Instructions for Univac 1108, Exec 8 Operating System

ORIGINAL PAGE IS
OF POOR QUALITY

```
LINKB5 TREE LKB5-(LKC1,LKC2,LKC3,LKC4,LKC5)
LINKB1 TREE LKB1
LINKB2 TREE LKB2
LINKB3 TREE LKB3
LINKB4 TREE LKB4
ROOT TREE LEG00-(LINKB1,LINKB2,LINKB3,LINKB4,LINKB5)
LEG00 INCLUDE MAIN,DRIVER,BLKDAT,
LKB1 INCLUDE PLUMIN,INITP,GASRD,GASTAP,PARTIN,LIFIN,WOFA,AOASTF,SETNTC,P
,ARTPM,MASCOM,PLTOUT,STLINE,SPACET
LKB2 INCLUDE TRANS,PUREX,ARASSL,SPECIE,ILUPOF,POP
LKB3 INCLUDE ELMPIN,IOMTOB,TOB,SPECIB,WALPRP
LKB4 INCLUDE PBLTPC,IDMPRO,PRO,IDMPDT,PDT,BLEXIT,PARLOK,INRSCP,TAPMOV,RF
,ADF,INTERP,TRACEP,GAS,START,PHEAD,WTFLOP,WRITP
LKB5 INCLUDE PHASE1
LKC1 INCLUDE STRNGR,SOKSOL,PHYSOL,CARCTR,SLPLIN,AVERAG,STGMOD
LKC2 INCLUDE MORSCK,PREFM,VMOD2,TURB,LIMITS,ITARM,FSHOCK,WEAK,DELTAF,INTR
,OP,OUT,SLSKIP,PREREC,FREERC,WTFLOP,OUTSTM,INTT
LKC3 INCLUDE THRUST,TLTSOF,MASSCK,ITERM,LIPEAK
LKC4 INCLUDE MCCRMK,COUNLA
LKC5 INCLUDE TRETPM,PRAUDT,HYPER,PARSTR,MOCSOL,FARINT,IRAD,PLOAD,SLIPT,N
,ORMCK,MESH,EXPCOR,OVEREX,FSHOCC,WEKK,FNTRPP,DELTEF
LEG01 GLOBAL MET,VTRY,PARTP3,INTCR,BOMOUT,FSTAL,PTEK,CHEMXX
LEG02 GLOBAL ELMDAT,LIPPT,SPRESW,PRAD,PCIC,VMIX3,CHEMCN,VISLX
LEG04 GLOBAL HUL,DRUG,CHEQ,VAROF,ISTRT,IPMX,OFSTAR,CAPUR,TRPRT,DRAGCF
LEG05 GLOBAL GRINT,RSTART,FAB,CPYUK,MOL,WRITET,TEMPER,WAFT,DISCON
LEG06 GLOBAL TAPRIT,XSICOM,XXSF,PSLD,TSPT,LAPPA,SIGNAL,NSF,FREE
LEG07 GLOBAL MASSC,WT,AUX,STEPC,TPEH,CLAT,CASCOM,FORCE,SKIPPY
LEG08 GLOBAL GASDAT,PARTP2,PARTP4,CUTEM,CRITER,IOL,ACOM,CONTEL
LEG09 GLOBAL VLIB,DATAR,SPEL,PARTP1,TFLAG,VARSL,PARTWT
LEG10 GLOBAL PDLL,VSON,PRNV
LKB5  GLOBAL CHEMYY,VMIX2,RUF,VMIX5,CHEMXY,PARSTU,AVPRP2,VMIX1,AVERDF
LKB5  GLOBAL POINTC,SLIPPT,LSAC,PUNEXT,IRN,LIPEX,WRITIT,ISEA,MASOUT,
LKB5  GLOBAL AMF,PUT,PSTR,VMIX6,VMIX4,EXPER,TEMPC3,PSEC,INTEU,OVERLA
LKB5  GLOBAL TOTAL,GLOBAL,DROP,TEMPO2,TEMPO1,CPSV,LSTRLN
  END
```

Fig. A-3 RAMP2F Loader Directives for the CDC Segmentation Loader

```
LINK3 TREE LINK4
LINK5 TREE LINK7
LINK1 TREE LINK2=(LINK3,LINK4,LINK5,LINK6)
ROOT TREE LEVEL=(LINK1,LEVEL2,LINK1)
LEVEL INCLUDE FLCOM,MAIN,MATSI,MATSI,INPUT,DIRECT,MAT-R,THERL,FAIRS,TERM
,L,ENF,ERP,TAYLOR,LIAPU,SLOPE,TRANCE,SRC...,CHOP,NBCMN,STATE,SECOND,TIME
,F,RE...Y,TOUT
LINK3 INCLUDE TOM,LIMAT,HISTXI,FIRSTS,DATE,SETUP
LINK7 INCLUDE SLOPE,PSEC...
LINK4 INCL... DISCIN,FCOM,PRCASE
LINK5 INCL... PUT
LINK6 INCL... DATE
LINK7 INCL... TLOT,MCLOSE,OPLE,ALL A...LY,ICOEFF,IMAP,LINCEL,ITERAT,N
,NNCEL,RECOUT,TRINT,PLMAS,FILOC,FILOT,INPLEG,POINTS,NEWIT,OUTPL,PLOT
LINKA INCLUDE EDITORP,LOGHSE,TAINOV,RLOOP,IDMTAD,TA...
LEGAL GLOBAL MINCOM,TAPECOM,CPLCOM,EPSCOM,CMLCOM,COECOM,STTCOM
LEGAL GLOBAL PRPCOM,PLPCOM,PUMCOM,FQ...,FLXC...,T,...COM,PLTCOM
LEGAL GLOBAL VALCOM,VARCOM,T...COM,...COM,...PRCOM,...ICOM,PRPLOM,ALTCOM
L...AL STATCOM,...C...,LSCOM,...ECOM,INTCOM,...COM
LINK2 GLOBAL SLPCOM
LINK7 GLOBAL FITCOM,...TCOM,PSLCOM,TRICOM,OUTCOM
  END
```

Fig. A-4 BLIMPJ Loader Directives for the CDC Segmentation Loader

Appendix B

INSTRUCTIONS FOR MODIFYING THE LOCKHEED METHOD OF CHARACTERISTICS PROGRAM TO USE A RAMP2F VISCOUS IDEAL GAS EXIT PLANE STARTLINE

Appendix B

One of the exit plane start line options available in the RAMP2 program is a startline that is compatible with the Lockheed Method of Characteristics Program (MOC)*. This startline assumes an ideal gas approximation of variable total temperature which occurs in viscous flows. The RAMP program will output this startline following a nozzle, boundary layer and exit plane startline run of the RAMP2F program. The startline and associated total conditions are punched for the exit plane of the nozzle. The axial coordinates are assumed to be zero and the radial coordinates of the startline are non-dimensionalized by the exit radius. The total conditions, $\bar{R}, \gamma$, T_o, P_o are mass flow averaged based on the exit plane conditions.

The subroutines in the MOC program which required modifications are: TABLE, POFEM, EMOFP, MOCSOL and PRANDT. Figures B-1 through B-5 present listings of the routines as they should be to utilize the RAMP2F ideal gas MOC startline. Changes to the program are noted on he listings using a * for statements that have been modified and a + for statements that have been added.

*Smith, S.D., and A.W. Ratliff, "User's Manual - Varialbe O/F Ratio Method of Characteristics Program for Nozzle and Plume Analysis," LMSC-HREC D162220-IV, Lockheed Missiles & Space Company, Huntsville, Ala., June 1971.

Table B-1

SUBROUTINE TABLE MODIFIED FOR USE WITH RAMP2 MOC STARTLINE

```
      SUBROUTINE TABLE(SS,VV,IF)
      THIS ROUTINE CALCULATES LOCAL GAS PROPERTIES FOR REAL OR IDEAL
      GAS AS A FUNCTION OF ENTROPY(SS) AND VELOCITY(VV)
      COMMON/ASICOM/XSIP(21,40)
      COMMON/CONTRL/IPUP(10),TCO(10)
      COMMON/DATAB/PHO(5,13C,2),IPHO(13C,2),APHO(1C9,2),
     *OFRAT(2C),STAE(2C,2),IVTAP(2,,2),TAB9(2550),
     *WALLCO(13C,5,2),IWALL(13C,2),ITRACS(1C5,2),IECHO(2)
      COMMON/GASCON/R,GAMMA,TO,PO,ASTER
      COMMON/CURA/IDEAL
      COMMON/FAR/T,P,SP
      COMMON/TEPPER/TR,CF
      DIMENSION VENTRY(5,2),VP(2)
C     AN ASTERIK (*) INDICATES GAS PROPERTIES OUTSIDE RANGE OF TABLE
C     A BLANK MEANS WITHIN RANGE OF TABLE
      DATA AA/4H*   /
      DATA BB/4H    /
      ASTER=BB
      IV=1
      IDEAL=0
      S=SS
      V=VV
      SP=S
      VP(1)=V
      VP(2)=V
C     FIND NUMBER OF ENTROPY CUTS
      IS=ICOM(1)
C     IF ONLY ONE ENTROPY CUT, YOU HAVE EITHER FROZEN OR IDEAL GAS
      IF(IS.EQ.1)GO TO 1CC
C     IF ENTROPY OUTSIDE RANGE OF TABLE,SET TO LOWEST OR HIGHEST VALUE
      IF(S.LT.STAB(IF,1)) SP=STAB(IF,1)
      IF(S.GT.STAB(IF,IS)) SP=STAB(IF,IS)
      IF(S.NE.SP) ASTER=AA
C     BRACKET INPUT ENTROPY BETWEEN THE (I) AND (I-1) TABLE VALUES
      DO 10 I=2,IS
      IF(SP.LE.STAB(IF,I))GO TO 2C
   10 CONTINUE
      I=IS
   2C JS=I-1+IV
C     SET VELOCITY TO LOWEST OR HIGHEST TABLE VALUE. IF OUTSIDE RANGE
      IF(V.LT.TAB(IF,JS,1,1)) VP(IV)=TAB(IF,JS,1,1)
      IVMAX=IVTAB(IF,JS)
      IF(V.GT.TAB(IF,JS,IVMAX,1)) VP(IV)=TAB(IF,JS,IVMAX,1)
C     BRACKET INPUT VELOCITY BETWEEN (L) AND (L-1) TABLE VALUES
      DO 3C L=2,IVMAX
      IF(VP(IV).LE.TAB(IF,JS,L,1)) GO TO 4G
   3C CONTINUE
      L=IVMAX
C     CALCULATE FRACTION OF WAY BETWEEN (L) AND (L-1) VELOCITY CUTS
   4G HW=(VP(IV)-TAB(IF,JS,L-1,1))/(TAB(IF,JS,L,1)-TAB(IF,JS,L-1,1))
C     SET LOCAL VELOCITY AT FIRST ENTROPY CUT (IV=1)
      VENTRY(1,IV)=VP(IV)
C     SET LOCAL GAS CONSTANT (R) AT FIRST ENTROPY CUT (IV=1)
      VENTRY(2,IV)=(1.-HW)*TAB(IF,JS,L-1,2)+HW*TAB(IF,JS,L,2)
C     SET LOCAL GAMMA AT FIRST ENTROPY CUT (IV=1)
      VENTRY(3,IV)=(1.-HW)*TAB(IF,JS,L-1,3)+HW*TAB(IF,JS,L,3)
C     XSIT IS A WEIGHTING FACTOR DEFINING A PATH BETWEEN TEMPERATURES
C     AT TWO VELOCITY CUTS
      XSIT=XSI(IF,JS,L-1,1)
C     XSIP IS SIMILAR TO XSIT BUT DEFINES A PATH BETWEEN PRESSURES
      XSIP=XSI(IF,JS,L-1,2)
C     SET LOCAL TOTAL TEMP AT FIRST ENTROPY CUT
      VENTRY(4,IV)=TAB(IF,JS,L-1,4)+(TAB(IF,JS,L-1,1)*TAB(IF,JS,L-2,1)/
     *TAB(IF,JS,L-2,2)-VENTRY(1,IV)*VENTRY(1,IV)/VENTRY(5,IV))*.5/XSIT
C     SET LOCAL TOTAL PRESSURE AT FIRST ENTROPY CUT
      VENTRY(5,IV)=TAB(IF,JS,L-2,5)*((VENTRY(4,IV)/TAB(IF,JS,L-1,4))
     ***XSIP)
```

Table B-1 - (Concluded)

```
C     VP WILL EQUAL 1 IF INPUT VELOCITY IS WITHIN RANGE OF TABLE
      IF(VP(IV).NE.1)GO TO 70
C     PERFORM CALCULATIONS IDENTICAL TO THOSE DESCRIBED. FOR SECOND
C     ENTROPY CUT, OR IF THIS IS ALREADY DONE GO TO 80 AND START
C     INTERPOLATING ON ENTROPY
   60 IF(IV.EQ.2) GO TO 80
      IV=2
      GO TO 20
C     IF OUTSIDE RANGE OF TABLE, CALCULATE GAS PROPERTIES FROM LAST
C     TABLE VALUES USING IDEAL GAS RELATIONS
   70 RHOVENTRY(2,IV)
      ASTER=AA
      GAMMA=VENTRY(3,IV)
      T=VENTRY(4,IV)
      P=VENTRY(5,IV)
      EM=VP(IV)/SQRT(R*GAMMA*T)
      TO=T*(1.+(GAMMA-1.)*EM**2/2.)
      PO=P*((TO/T)**(GAMMA/(GAMMA-1.)))
      VENTRY(4,IV)=TOFV(V)
      VENTRY(5,IV)=POFE**(EMOFV(V),...)
      GO TO 60
   80 IF(IS.EQ.1) GO TO 110
C     FIND FRACTION OF WAY BETWEEN (I) AND (I-1) ENTROPY CUTS
      WS=(SP-STAB(IF,I-1))/(STAB(IF,I)-STAB(IF,I-1))
C     FIND R,GAMMA,T BY LINEAR INTERPOLATION BETWEEN ENTROPY CUTS
      R=VENTRY(2,1)+WS*(VENTRY(2,2)-VENTRY(2,1))
      GAMMA=VENTRY(3,1)+WS*(VENTRY(3,2)-VENTRY(3,1))
      T=VENTRY(4,1)+WS*(VENTRY(4,2)-VENTRY(4,1))
C     CALCULATE WEIGHTING FUNCTION (ANALOGOUS TO XSIT AND XSIP)
      WSP=(SP/R-STAB(IF,I-1)/VENTRY(2,1))/(STAB(IF,I)/VENTRY(2,2)-
     *STAB(IF,I-1)/VENTRY(2,1))
C     FIND PRESSURE USING WEIGHTING FUNCTION (WSP)
      P=VENTRY(5,1)*EXP(WSP*ALOG(VENTRY(5,2)/VENTRY(5,1)))
C     CALCULATE MACH NO
      EM=V/SQRT(GAMMA*R*T)
C     CALCULATE STAGNATION CONDITIONS
      TO=T*(1.+(GAMMA-1.)*EM**2/2.)
      PO=P*((TO/T)**(GAMMA/(GAMMA-1.)))*EXP(SP/R)
      GO TO 200
  100 IVMAX=IVTYPE(IF,1)
C     IF IDEAL, GO TO 120 AND SET GAS PROPERTIES. IF REAL, GO BACK
C     AND INTERPOLATE BETWEEN VELOCITIES
      IF(IVMAX.EQ.1) GO TO 120
      SP=STAB(IF,1)
      I=1
      IV=2
      GO TO 20
C     SET GAS PROPERTIES HERE WHEN ONLY ONE ENTROPY CUT (IS = 1)
  110 GAMMA=VENTRY(3,2)
      R=VENTRY(2,2)
      T=VENTRY(4,2)
      P=VENTRY(5,2)
      EM=V/SQRT(GAMMA*R*T)
      TO=T*(1.+(GAMMA-1.)*EM**2/2.)
      PO=VENTRY(5,2)*((TO/T)**(GAMMA/(GAMMA-1.)))*EXP(SP/R)
      GO TO 200
C     SET GAS PROPERTIES HERE WHEN IDEAL GAS
  120 R=TAB(IF,1,1,2)
      GAMMA=TAB(IF,1,1,3)
      TO=TSP(IF,1,1,4)
      PO=TAB(IF,1,1,5)
      IDEAL=1
      CP=GAMMA*R/(GAMMA-1.)
      TR=TO
      TO=OF/CP
      T=TOFV(VV)
  200 CONTINUE
      RETURN
      END
```

Table B-2

SUBROUTINE ROFEM MODIFIED FOR USE WITH RAMP2 MOC STARTLINE

```
C     PRESSURE AS FUNCTION OF MACH NO.(FM) AND ENTROPY(S)
      FUNCTION POFEM(FM,S)
      COMMON/GASCON/R,GAMMA,TO,PO,ASTER
      COMMON/TEMPER/TR,DF
      T1=TO
      T2=TR
      GM102=(GAMMA-1.)*.5
      RATGAM=GAMMA/(GAMMA-1.)
C     COMPUTE LOCAL TOTAL PRESSURE (PSTAR) AS A FUNCTION OF
C     HEAD LOSS (EXP(S/R))
      PSTAR=PO/EXP(S/R)*((T1/T2))**RATGAM
C     COMPUTE LOCAL STATIC PRESSURE
      POFEM=PSTAR/((1.+GM102*FM**2)**RATGAM)
      RETURN
      END
```

Table B-3

SUBROUTINE EMOFD MODIFIED FOR USE WITH RAMP2 MOC STARTLINE

```
      FUNCTION EMOFP(P,S)
C     MACH NO. AS FUNCTION OF PRESSURE(P) AND ENTROPY(S)
      COMMON/GASCON/R,GAMMA,TO,PO,ASTER
     1COMMON/TEMPER/TR,DF
      RATGAM=GAMMA/(GAMMA-1.)
      GM102=(GAMMA-1.)*.5
C     COMPUTE LOCAL TOTAL PRESSURE AS A FUNCTION OF HEAD LOSS
      PSTAR=PO/EXP(S/R)*((TO/TR))**RATGAM
C     COMPUTE LOCAL MACH NUMBER FROM THERMALLY PERFECT GAS RELATIONS
      EMOFP=SQRT(((PSTAR/P)**(1./RATGAM)-1.)/GM102)
      RETURN
      END
```

Table B-4

SUBROUTINE MOCSOL MODIFIED FOR USE WITH RAMP2F MOC STARTLINE

```
      SUBROUTINEMOCSOL(IN,KN,IN1,KN1,IN2,KN2,IFLAG,ITYPE)
C     MOCSOL SOLVES THE CHARACTERISTIC EQUATIONS AT THE NEW POINT
C     (IN,KN) DESCRIBES THE NEW POINT
C     (IN1,KN1,IN2,KN2) DESCRIBES THE TWO KNOWN POINTS
C     (IFLAG) INDICATES IF AN ERROR EXISTS IN THE CALCULATION
C     (ITYPE) INDICATES THE TYPE OF CALCULATION TO BE MADE
C**** ITYPE=11 COMPUTES INTERIOR POINT
C       21 COMPUTES LOWER WALL
C       22 COMPUTES UPPER WALL
C       31 COMPUTES LOWER FREE BOUNDARY
C       32 COMPUTES UPPER FREE BOUNDARY
      COMMON/CONTROL/IRUN(10),ICON(16)
      COMMON/CRITER/CONVRS(10),ITLIM(10)
      COMMON/DATAR/PHO(8,100,2),IPHO(100,2),APHO(100,2),
     *OFRAT(20),STAB(20,2),IVTAB(20,2),TABB(2600),
     *WALLCO(100,6,2),IWALL(100,2),ITPARS(100,2),IGROW(2)
      COMMON/GASCON/R,GAMMA,TO,PO,ASTER
      COMMON/FIXITE/IFREE,TINF,OF3
      DATA PI04/.78539816/
C     TRUNCATE OFF SECOND DIGIT AND RETAIN FIRST (1-INTERRIOR
C     2-WALL,3-FREE BOUND)
      ITYPE1=ITYPE/10
C     FILTER OUT AND RETAIN SECOND DIGIT (1-INTERRIOR OR LOWER,2-UPPER)
      ITYPE2=ITYPE-(ITYPE/10)*10
C     SET SYMBOLS FOR NEW POINT
      I=IN
      K=KN
C     SET SYMBOLS FOR TWO KNOWN POINTS
      I1=IN1
      K1=KN1
      I2=IN2
      K2=KN2
C     SEE IF AXISYMMETRIC OR TWO DIMENSIONAL
      AXO2D=ICON(7)
C     INITIALIZE CONTROL CONSTANTS
      ITERAT=1
      IFLAG=1
      IPASS=1
C     FIND ALL NECESSARY CHARACTERISTIC DATA AT FIRST KNOWN POINT
      R1=PHO(1,I1,K1)
      X1=PHO(2,I1,K1)
      V1=PHO(3,I1,K1)
      THETA1=PHO(4,I1,K1)
      S1=PHO(5,I1,K1)
      U1=PHO(6,I1,K1)
      OF1=PHO(8,I1,K1)
      CALL FABLE(OF1,S1,V1)
      GAMMA1=GAMMA
      RC1=R
      P1=PDFEM(EMOFV(V1),S1)
C     FIND ALL NECESSARY CHARACTERISTIC DATA AT SECOND KNOWN POINT
      R2=PHO(1,I2,K2)
      X2=PHO(2,I2,K2)
      V2=PHO(3,I2,K2)
      THETA2=PHO(4,I2,K2)
      S2=PHO(5,I2,K2)
      U2=PHO(6,I2,K2)
      OF2=PHO(8,I2,K2)
      CALL FABLE(OF2,S2,V2)
      GAMMA2=GAMMA
      RC2=R
      P2=PDFEM(EMOFV(V2),S2)
C     AVERAGE KNOWN VALUES FOR INITIAL GUESS AT NEW POINT VALUES
      V3=(V1+V2)*.5
      THETA3=(THETA1+THETA2)*.5
      S3=(S1+S2)*.5
      OF3=(OF1+OF2)*.5
      U3=(U1+U2)*.5
```

Table B-4 (Continued)

```
C     GUESS INITIAL AVERAGE VALUES
      U1B=U1
      U2B=U2
      R1B = (R1+R2)*.25
      R2B = (R1+R2)*.25
      THET1B=THETA1
      THET2B=THETA2
      V2B=V2
      V1B=V1
      GOTO(10,15,15),ITYPE1
C     THIS IS FOR INTERRIOR OR WALL POINT
   10 CONVM1=(V1+V2)*.5
      ITLIMT=ITLIM(1)
      CONVAL=CONVRG(1)
      GO TO 20
C     THIS IS FOR FREE BOUNDARY POINT
   15 CONVM1=(THETA1+THETA2)*.5
      ITLIMT=ITLIM(2)
      CONVAL=CONVRG(2)
      IF(IFREE.EQ.2)CONVAL=CONVAL/2.
   20 GOTO(25,30,30),ITYPE1
C     CALCULATE LEFT AND RIGHT RUNNING CHARACTERISTIC ANGLE FOR
C     INTERRIOR POINT
   25 BETA1=THET1B-U1B
      BETA2=THET2B+V2B
      GOTO45
   30 GOTO(35,40),ITYPE2
C     CALCULATE CHARACTERISTIC ANGLES FOR LOWER BOUNDARY
   35 BETA1=THET1B-U1B
      BETA2=THET2B
      GOTO45
C     CALCULATE CHARACTERISTIC ANGLES FOR UPPER BOUNDARY
   40 BETA1=THET1B
      BETA2=THET2B+V2B
C     FIND INTERSECTION OF LEFT AND RIGHT RUNNING CHARACTERISTIC LINES
C     FROM TWO KNOWN POINTS. THIS IS LOCATION OF NEW POINT
   45 IF(BETA1.EQ.BETA2) GO TO 161
      CALL INPSCT(R1,X1,BETA1,R2,X2,BETA2,R3,X3)
      GOTO(55,50,55),ITYPE1
C     SOLVE BOUNDARY EQUATION FOR R AND THETA AT INPUT X
   50 CALL BOUNC(R3,X3,THET3,ITYPE2)
C     LOAD IN NEW COORDINATES
   55 PHO(I,J,K)=R3
C     ICON(7)=1-TWO DIMENSIONAL, ICON(7)=2-AXISYMMETRIC
      PHO(2,J,K)=X3
      IF(ICON(7).EQ.1.AND.R3/(R1+R2).LT.0.) GO TO 165
C     IPASS=1-FIRST TIME THROUGH,IPASS=2 -SUBSEQUENT PASSES
      GOTO(60,65),IPASS
C     CHECK ON ORIENTATION OF CHARACTERISTIC LINES FIRST TIME THROUGH
   60 IQUAD1=1
      IF(ABS(BETA1).LE.PIO4)IQUAD1=2
      IQUAD2=1
      IF(ABS(BETA2).LE.PIO4)IQUAD2=2
      DELTA1=1-2*(IQUAD1/2)
      DELTA2=1-2*(IQUAD2/2)
      IPASS=2
C     CALCULATE ROTATIONAL TERM OF COMPATABILITY EQN. (EQNS 6-29 REF 1)
   65 F1=ROTERM(THET1B,DELTA1,-U1B,PHO(IQUAD1,J,K),PHO(IQUAD1,I1,K1))
      F2=ROTERM(THET2B,DELTA2,U2B,PHO(IQUAD2,I,K),PHO(IQUAD2,I2,K2))
      GO TO (71,75,75),ITYPE1
   71 S3=S1+F1*(S2-S1)/(F1+F2)
      OF3=OF1+F1*(OF2-OF1)/(F1+F2)
      GOTO90
   75 GOTO(80,85),ITYPE2
C     ASSUME ENTROPY AND VELOCITY CONSTANT ALONG LOWER BOUND
   80 S3=S2
      OF3=OF2
      V3=V2
      GOTO86
```

Table B-4 (Continued)

```
C     ASSUME ENTROPY AND VELOCITY CONSTANT ALONG UPPER BOUND
   85 S3=S1
      OF3=OF1
      IF(ITER#1.EQ.1)V3=V1
   86 GOTO(90,90,87),ITYPE1
   87 FB=1+2*(ITYPE2-2)
      IEQ=IEQNO(ITYPE2)
C     GET A WEIGHTED NEW VELOCITY FROM PRESSURE WHEN FREE BOUNDARY
      V3=VOFEM(FSMOFP(FNUTN(THETA3,X3,R3,ITYPE2),OF3,S3))*.5+V3*.5
   90 SU1B=SIN(L1B)
C     CALCULATE AVERAGE VALUES OVER STEP LENGTH
      R1B = (R1+R3)*.5
      R2B = (R2+R3)*.5
      CU1B=COS(L1B)
      SU2B=SIN(L2B)
      CU2B=COS(L2B)
C     EQUATION 6-25 REF 1
      Q1=CU1B/(SU1B*V1B)
      Q2=CU2B/(SU2B*V2B)
      CALL FABLE(OF3,S3,V3)
      P3=POFEM(EMOFV(V3),S3)
      RAV1=(RC1+R)*.5
      RAV2=(RC2+R)*.5
      GAMAV1=(GAMMA1+GAMMA)*.5
      GAMAV2=(GAMMA2+GAMMA)*.5
C     EQUATION 6-25 REF 1
      OFBAR=0.5*(OF1+OF3)
      SBAR=0.5*(S1+S3)
      CALLFABLE(OFBAR,SBAR,V1B)
      TAV1=TOFV(V1B)
      RHOAV1=RHOFEM(EMOFV(V1B),SBAR)
      OFBAR=0.5*(OF2+OF3)
      SBAR=0.5*(S2+S3)
      CALLFABLE(OFBAR,SBAR,V2B)
      TAV2=TOFV(V2B)
      RHOAV2=RHOFEM(EMOFV(V2B),SBAR)
      DXSI1=S3-S1
      DXSI2=S3-S2
      CI1=CU1B/SU1B/(V1B*V1B)*(OF3-OF1)
      CI2=CU2B/SU2B/(V2B*V2B)*(OF3-OF2)
      B1=SU1B*CL1B*DXSI1/(RAV1*GAMAV1)
      B2=SU2B*CL2B*DXSI2/(RAV2*GAMAV2)
C     EQUATION 6-25 REF 1
      G1=0.0
      G2=0.0
      IF(ABS(Q1B).GT.0.0)G1=SIN(THET1B)*F1/R1B
      IF(ABS(Q2B).GT.0.0)G2=SIN(THET2B)*F2/R2B
      GOTO(95,100,115),ITYPE1
C     CALCULATE NEW VELOCITY FOR INTERRIOR PT.(EQN 6-19 REF 1)
   95 V3=(THETA1-THETA2+Q1*V1+Q2*V2+AXO2D*(G1+G2)-B1-B2)/(Q1+Q2)
     **(CI1+CI2)/(Q1+Q2)
C     CALCULATE NEW FLOW ANGLE FOR INTERRIOR PT.(EQN 6-20 REF 1)
      THETA3=THETA2+G2*(V3-V2)-AXO2D*G2+B2
     *-CI2
      GOTO130
  100 GOTO(105,110),ITYPE2
C     CALCULATE NEW VELOCITY FOR LOWER WALL(EQN 6-23 REF 1)
  105 V3=(THETA1-THETA3+G1*AXO2D-B1+Q1*V1)/Q1
     **CI1/G2
      GOTO130
C     CALCULATE NEW VELOCITY FOR UPPER WALL(EQN 6-21 REF 1)
  110 V3=(THETA2-THETA2+G2*AXO2D-B2+Q2*V2)/Q2
     **CI2/Q2
      GOTO130
  115 GOTO(120,125),ITYPE2
C     CALCULATE NEW FLOW ANGLE FOR LOWER FREE BOUND (EQN 6-24 REF 1)
  120 THETA3=THETA1-G1*(V3-V1)+G1*AXO2D-B1
     **CI1
      GOTO130
C     CALCUALTE NEW FLOW ANGLE FOR UPPER FREE BOUND (EQN 6-22 REF 1)
  125 THETA3=THETA2+Q2*(V3-V2)-G2*AXO2D+B2
     *-CI2
```

Table B-4 (Concluded)

```
  130 CALL FABLE(OF3,S3,V3)
C     CALCULATE NEW MACH ANGLE
      U3=UOFV(V3)
      GOTO(135,135,140),ITYPE1
C     SET CONVERGENCE CRITERIA
  135 CONV=V3
      GOTO145
  140 CONV=THETA3
C     CHECK FOR CONVERGENCE
  145 IF((ABS ((CONV+3.14)/(CONVM1+3.14)-1.)-CONVAL).LE.0.)GO TO 165
      IF((ITERAT-ITLIMT).GT.0)GO TO 160
C     RE CALCULATE AVERAGE VALUES AND INCREMENT FOR NEXT PASS
      V1B=(V1+V3)*.5
      V2B=(V2+V3)*.5
      U1B=(U1+U3)*.5
      U2B=(U2+U3)*.5
      THET1B=(THETA1+THETA3)*.5
      THET2B=(THETA2+THETA3)*.5
      ITERAT=ITERAT+1
      GOTO(150,150,155),ITYPE1
  150 CONVM1=V3
      GOTO20
  155 CONVM1=THETA3
      GOTO20
  160 CALL ERRORS(12)
C     ERRORS(12) STATES - MOCSOL UNC
      IFLAG=2
      GOTO170
  161 CALL ERRORS(23)
      IFLAG=2
      GO TO 170
  165 BETA1=(THETA1+THETA3-U1-U3)*.5
      BETA2=(THETA2+THETA3+U2+U3)*.5
      IF(ITYPE1.EQ.1)CALLINPSCT(R1,X1,BETA1,R2,X2,BETA2,PHO(1,I,K),PHO(
     12,I,K))
C     LOAD FINAL CHARACTERISTIC DATA INTO PHO ARRAY WHEN CONVERGED
      PHO(3,I,K)=V3
      PHO(4,I,K)=THETA3
      PHO(5,I,K)=S3
      PHO(6,I,K)=U3
      PHO(7,I,K)=0.
      PHO(8,I,K)=OF3
      IPHO(I,K)=ITYPE1
      APHO(I,K)=ASTER
  170 RETURN
      END
```

Table B-5

SUBROUTINE PRANDT MODIFIED FOR USE WITH RAMP2 MOC STARTLINE

```
      SUBROUTINE PRANDT(I,J,THETAB,NPM,IFLAG,ITYPE)
C     THIS ROUTINE COMPUTES PRANDTL-MEYER EXPANSION ANGLE AND
C     DIVIDES INTO SERIES OF EXPANSION RAYS
C     (I) - IDENTIFIES CORNER POINT
C     (J) - IDENTIFIES A CHARACTERISTIC LINE
C     (THETAB) - BOUNDARY ANGLE
C     (NPM) - NUMBER OF INCREMENTS
C     (IFLAG)  - ERROR FLAG
C     (ITYPE) - IDENTIFIES UPPER (2) OR LOWER(1) BOUNDARY
      COMMON/CONTRL/IRUN(10),ICON(16)
      COMMON/TESTV/IVTEST
      COMMON/CRITER/CONVRG(10),ITLIM(10)
      COMMON/DATA2/PHO(8,100,2),IPHO(100,2),APHO(100,2),
     *OFRAT(20),STAB(20,2),IVTAB(20,2),TAB(2500),
     *WALLCO(100,6,2),IWALL(100,2),ITRANS(100,2),IEONOW(2)
      COMMON/GASCON/R,GAMMA,TO,PO,ASTER
      COMMON/STEPC/STEP(10)
      IFLAG=1
      ITYPE1=ITYPE-(ITYPE/10)*10
C     SET CORNER  CONDITIONS
      RC=PHO(1,I,J)
      XC=PHO(2,I,J)
      SC=PHO(5,I,J)
      OF=PHO(8,I,J)
      CALL TABLE(OF,SC,PHO(3,I,J))
      VC=PHO(3,I,J)
      THETAC=PHO(4,I,J)
      NPM=ICON(16)
      IF(NPM.NE.0) GO TO 5
C     CALCULATE NUMBER OF INCREMENTS
      NPM=ABS((THETAB-THETAC)/STEP(1))
      NPM=NPM+1
      IF(ITYPE1.EQ.1.AND.NPM.GT.90) NPM=90
      IF(ITYPE1.EQ.2.AND.NPM.GT.90)NPM=90
C     INITIALIZE
    5 CONTINUE
      VI=VC
      THETAI=THETAC
      K=0
      DPI02 = 1.5707963/FLOAT(NPM)
      ARG = 0.
   10 K=K+1
      ARG = ARG+DPI02
C     CALCULATE WEIGHTING FUNCTION(LOOSENS INITIAL RAYS,TIGHTENS FINAL)
      DEGPMP=(THETAB-THETAC)*(ARG/1.5707963)-(THETAI-THETAC)
      KPI=K+1
C     HOLD CORNER COORDINATES(RC,XC) AND ENTROPY (SC) CONSTANT
C     THROUGHOUT EXPANSION
      PHO(1,KPI,J)=RC
      PHO(2,KPI,J)=XC
      PHO(5,KPI,J)=SC
      PHO(8,KPI,J)=OF
C     FIND VELOCITY(VSTAR) DOWNSTREAM OF EXPANSION RAY
      CALL THETFM(OF,SC,DEGPMP,VSTAR,VI,1,ITYPE)
      THETAI=THETAI+DEGPMP
C     SET CONDITIONS DOWNSTREAM OF EXPANSION
   30 PHO(3,KPI,J)=VSTAR
      PHO(4,KPI,J)=THETAI
      PHO(6,KPI,J)=UOFV(VSTAR)
      PHO(7,KPI,J)=0.
      IPHO(KPI,J)=5
      APHO(KPI,J)=ASTER
      VI=VSTAR
      IF(IVTEST.GT.0)NPM=K
      IF(IVTEST.GT.0)IVTEST=0
      IF(K.LT.NPM)GOTO10
      GOTO200
  200 RETURN
      END
```